

# UD2: Lenguajes de estructura para web: HTML y XHTML

Lenguajes de  
marcas y sistemas  
de gestión de  
información

# La World Wide Web (WWW)

- Conjunto de documentos residentes en diferentes máquinas, relacionados entre sí por enlaces.
- El lenguaje usado por el navegador para mostrar estos documentos es un lenguaje de marcas llamado HTML (*HyperText Markup Language*), combinado con hojas de estilo CSS para modificar el aspecto visual.
- ¿Qué diferencias tenemos con las aplicaciones de escritorio?
- Tanto HTML como el resto de las tecnologías web están en constante evolución. Es importante ser conscientes de ello y actualizar nuestros conocimientos.
- El éxito de HTML radica en su sencilla estructura y su sintaxis simple.

# Publicación de una página web

- Opción 1: Inserción en un servidor web propio.
- Opción 2: *Hosting*.



# Publicación de una página web

http://

Protocolo

[www.lawebdemiguel.com](http://www.lawebdemiguel.com)

Nombre del sitio web

/

Ruta de la raíz

index.htm

Documento HTML

# El navegador

- Es la herramienta o programa que es capaz de interpretar el código HTML escrito en la página web.
- El motor de renderizado del navegador web es la parte del navegador encargada de cargar el contenido (HTML, XML, imágenes, etc.) y la información del formato (CSS) y crear la representación visual de la página.
- Antes de 1994 los navegadores incluían pocas funcionalidades.
- En 1994 aparece Netscape, con extensiones HTML **propias** que amplían las opciones de visualización.
- Microsoft publica Internet Explorer, dando comienzo a la llamada “Guerra de los navegadores”.
- En 1994 se crea la W3C (World Wide Web Consortium), con el objetivo de poner orden y crear estándares comunes a los diferentes navegadores.
- Posteriormente, las webs estáticas fueron dejando paso a webs dinámicas.

# Estructura y sintaxis HTML

- Los elementos que forman las páginas web están formados por etiquetas, que rodean todo el contenido de la página (texto, imágenes, etc.). Las etiquetas están formadas por un nombre, encapsulado por los símbolos < y >.
- HTML no distingue entre mayúsculas y minúsculas, con lo que podemos escribir el nombre de las etiquetas como queramos....
- .... pero es recomendable ser consistente, y en general, se acepta como consenso que deberían estar íntegramente en minúsculas.

# Estructura y sintaxis HTML

- Existen dos tipos de etiquetas, unas de apertura y otras de cierre.
- Un contenido debe estar rodeado por ambas etiquetas (la etiqueta de apertura antes del contenido y la de cierre después de éste).
- La etiqueta de apertura se diferencia de la de cierre en que ésta incluye el símbolo / antes del nombre de la etiqueta.
- Ambas etiquetas deben tener el mismo nombre.

`<nombre>` Contenido `</nombre>`

- Al conjunto de etiqueta de apertura, contenido y cierre se le denomina **elemento**.

# Estructura y sintaxis HTML

- En HTML no todas las etiquetas necesitan de apertura y cierre, ya que algunas no actúan como contenedoras. Un ejemplo de este tipo de etiquetas es `<hr>`, que dibuja una línea divisoria horizontal.
- Dentro de un elemento se pueden añadir **atributos**, que nos permiten caracterizarlo. La sintaxis de los atributos consiste en indicar dentro de la etiqueta de apertura el nombre del atributo, seguido por el símbolo = y el valor del atributo rodeado por comillas, generalmente dobles. Podemos indicar más de un atributo en una etiqueta.

`<nombre atributo1="valor1" atributo2="valor2">`



# Estructura y sintaxis HTML

- Anidamiento: el contenido de una etiqueta puede ser también otra etiqueta. Encerrar etiquetas dentro de otras es lo que se conoce como **anidamiento**.
- No se permite que las etiquetas se solapen entre sí.

```
<etiqueta1>  
    <etiqueta2>  
        <etiqueta3>  
        </etiqueta3>  
    </etiqueta2>  
</etiqueta1>
```

Anidamiento correcto

```
<etiqueta1>  
    <etiqueta2>  
        <etiqueta3>  
        </etiqueta2>  
    </etiqueta3>  
</etiqueta1>
```

Anidamiento incorrecto

# Estructura y sintaxis HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>
      Lenguajes de marcas
    </title>
  </head>
  <body>
    <h1>Bienvenidos a la página web de LiMa</h1>
    <p> Esta es la página web del módulo...</p>
  </body>
</html>
```

# Entidades HTML

- Según las normas que hemos visto hasta ahora, podríamos tener problemas a la hora de introducir cierto contenido.
- ¿Cómo interpreta el navegador un símbolo < como contenido de una etiqueta?
- Para solucionar esto, usamos las entidades, que son una construcción que permiten indicar al navegador el símbolo exacto que queremos representar sin usar el propio símbolo en el HTML.
- Siempre empiezan por & y terminan por ;

# Entidades HTML

Símbolo a representar	Entidad
<	&lt;
>	&gt;
'	&apos;
"	&quot;
&	&amp;
[Espacio en blanco]	&nbsp;

- ¿Por qué es necesaria una entidad para el espacio en blanco?

# Comentarios

- De forma general, un comentario es un texto incluido en el código fuente de un programa que será ignorado por el compilador o el intérprete cuando se compile/interprete el programa.
- En HTML es un texto que será ignorado por el navegador cuando lea el documento HTML. Su función es documentar partes de código, ya sea para uno mismo o para otros programadores.
- En HTML se indica de la siguiente forma:

```
<!-- esto de aquí es un comentario -->
```

```
<!-- esto es otro comentario -->
```

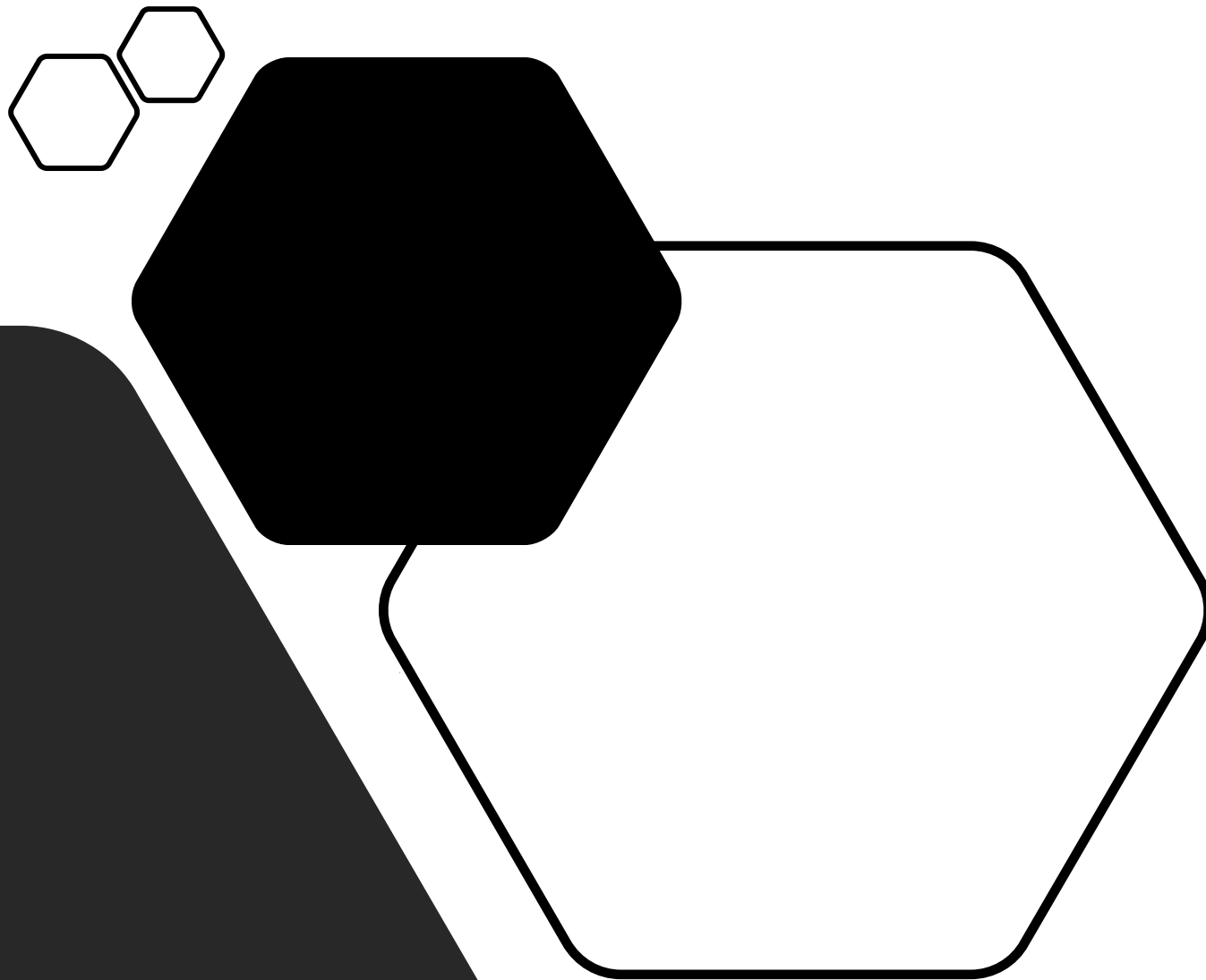
# Etiquetas HTML y atributos

- Existen muchas etiquetas HTML diferentes, en esta unidad veremos las más importantes.
- Existen atributos que pueden usarse en cualquier etiqueta, a los que llamaremos atributos globales.
- La mayor parte de los atributos, sin embargo, son específicos de una etiqueta o grupo de etiquetas.

# Atributos globales más empleados

- **class** (asigna una clase a una etiqueta. Varias etiquetas pueden tener el mismo valor en su atributo class).
- **id** (sirve para asignar manualmente un identificador a una etiqueta. Por definición, no puede haber dos etiquetas con el mismo id).
- **style** (sirve para aplicar estilos CSS en línea).
- **lang** (se usa para indicar el idioma en que está el contenido de la etiqueta).
- **hidden** (cuando se aplica, el contenido de la etiqueta no se muestra)
- **contenteditable** (permite que el contenido de la etiqueta se pueda editar desde el navegador)

# Etiquetas HTML





# <!DOCTYPE html>

- Indica que el documento es de tipo HTML, en su 5ª versión.
- Debe estar siempre presente.
- Debe ser siempre la primera línea de nuestro documento HTML.
- No tiene etiqueta de cierre.
- No se indican más atributos.

# <html>

- Debe estar siempre presente.
- Es la etiqueta raíz del documento.
- Todos los elementos del documento están contenidos, de forma directa o indirecta, dentro de <html>.
- Sus hijos directos son <head> y <body>

# <head>

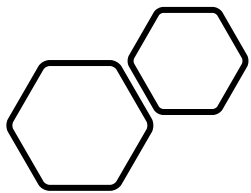
- Delimita la primera de las secciones de una página web.
- Es una parte **no visible** de la página, pero tiene efecto directo en la representación del resto del contenido.
- Una de las etiquetas que contiene es **<title>**, que se usa para asignar el título a una página web. Será también el nombre por defecto que se usará cuando se guarde la página web en los favoritos del navegador y además es uno de los términos emplean los buscadores web para posicionar la página, por lo que una buena elección de título es importante de cara a conseguir visitas.

# <head>

- Otra de las etiquetas contenidas en head es `<meta>`, que se usa para indicar los metadatos de la página. Dentro de `<meta>` nos interesa especialmente el atributo `charset`, que nos permite indicar el conjunto de caracteres que hemos usado para escribir el HTML, con la intención de que el navegador los interprete correctamente.
- `<meta>` tiene muchos otros atributos interesante posibles, que veremos más a fondo cuando hayamos creado unos cuantos documentos HTML.
- `<link>` se usa dentro de head para varios propósitos, por ejemplo, para vincular una hoja CSS externa o para asignar un icono a la pestaña

# <body>

- `<body>` delimita la sección del documento que contiene los elementos visibles del mismo.
- Existen decenas de etiquetas posibles dentro de `<body>`. Nos centraremos en las más usadas.
- Podemos clasificar todas las etiquetas contenidas en `<body>` en dos grandes grupos:
  - Etiquetas de bloque:
    - Ocupan todo el espacio horizontal de su contenedor (etiqueta padre).
    - Permiten contener otras etiquetas, tanto de bloque como de línea.
    - Introducen un cambio de línea al cerrarse.
  - Etiquetas de línea:
    - Ocupan solo el espacio horizontal de su contenido.
    - Contienen texto u otros elementos en línea.
    - No introducen un cambio de línea al cerrar de la etiqueta.



# Etiquetas de bloque

# Etiquetas de bloque: párrafos

- Con la etiqueta `<p>` indicamos un párrafo.
- Como etiqueta de bloque, introduce un salto de línea después del cierre de la etiqueta.
- Cuando queremos introducir texto directamente en el cuerpo del documento HTML deberíamos considerarlo como un párrafo.

# Etiquetas de bloque: párrafos

```
<!DOCTYPE html>

<html>
  <head>
    <meta charset="utf-8"/>
    <title>Párrafos HTML</title>
  </head>
  <body>
    <p>Este texto es un párrafo</p>
    <p>Este texto es otro párrafo, como puedes
    observar, se ha producido un cambio de
    línea entre ambos</p>
  </body>
</html>
```

Este texto es un párrafo

Este texto es otro párrafo, como puedes observar, se ha producido un cambio de línea entre ambos



# Etiquetas de bloque: cabeceras

- Etiquetas `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>` y `<h6>`.
- Sirven para indicar la cabecera de una sección.
- Están ordenados, de forma que `<h1>` debería usarse para indicar el título de una sección, `<h2>` el de una subsección, `<h3>` el de una subsección de menor nivel, etc.
- A partir de `<h4>` comienzan a ser más pequeñas que el texto estándar, pero podríamos usar CSS para modificar esto.

# Etiquetas de bloque: cabeceras

```
<!DOCTYPE html>

<html>
  <head>
    <meta charset="utf-8"/>
    <title>Cabeceras HTML</title>
  </head>
  <body>
    <h1>Cabecera h1</h1>
    <h2>Cabecera h2</h2>
    <h3>Cabecera h3</h3>
    <h4>Cabecera h4</h4>
    <h5>Cabecera h5</h5>
    <h6>Cabecera h6</h6>
  </body>
</html>
```

**Cabecera h1**

**Cabecera h2**

**Cabecera h3**

**Cabecera h4**

**Cabecera h5**

**Cabecera h6**

# Etiquetas de bloque: listas

- Etiquetas `<ol>`, `<ul>` y `<li>`:
  - `<ol>` se usa para declarar una lista ordenada.
  - `<ul>` se usa para declarar una lista no ordenada.
  - `<li>` indica cada uno de los ítems de una lista, ya sea ordenada o no ordenada.
  - Para que los elementos formen parte de la lista, además de ir dentro de `<ol>` o `<ul>` tienen que usar la etiqueta `<li>`. En caso contrario están dentro de la lista, pero no forman parte de ella, y no obtendrán numeración/viñeta.

# Etiquetas de bloque: listas

```
<h1>Listas en HTML</h1>
<h2>Lista ordenada:</h2>
<ol>
  <li>Primer item</li>
  <li>Segundo item</li>
  <li>Tercer item</li>
</ol>
<h2>Lista no ordenada:</h2>
<ul>
  <li>Un item</li>
  <li>Otro item</li>
  <li>Ultimo item</li>
</ul>
```

## Listas en HTML

### Lista ordenada:

1. Primer item
2. Segundo item
3. Tercer item

### Lista no ordenada:

- Un item
- Otro item
- Ultimo item

# Etiquetas de bloque: listas de definición

- Etiquetas `<dl>`, `<dt>` y `<dd>`:
  - `<dl>` declara una lista de definiciones, en la que vamos a tener, por una parte, un término (`<dt>`), y por otra su correspondiente definición (`<dd>`).
  - En las listas de definición usamos parejas de `<dt>` y `<dd>`, no usamos la etiqueta `<li>`.

# Etiquetas de bloque: listas de definición

```
<!DOCTYPE html>

<html>
  <head>
    <meta charset="utf-8"/>
    <title>Listas de definición con HTML</title>
  </head>
  <body>
    <dl>
      <dt>HTML</dt>
      <dd>Lenguaje de marcas muy fácil de aprender</dd>
    </dl>
    <dl>
      <dt>CSS</dt>
      <dd>Lenguaje para aplicar estilos a HTML</dd>
    </dl>
  </body>
</html>
```

HTML

Lenguaje de marcas muy fácil de aprender

CSS

Lenguaje para aplicar estilos a HTML

# Etiquetas de bloque: datos de contacto

- Etiqueta `<address>`.
- Representa información de contacto: una dirección postal, número de teléfono, correo electrónico, etc.
- Representación gráfica: el texto estará en cursiva.
- La información de contacto debería estar asociada a la etiqueta que contiene a `<address>` o bien representa al cuerpo de la página (`<body>`).

# Etiquetas de bloque: datos de contacto

```
<!DOCTYPE html>

<html>
  <head>
    <meta charset="utf-8"/>
    <title>Información de contacto con HTML
    </title>
  </head>
  <body>
    <address>Puedes contactar con el centro
    educativo en el teléfono 886 151 924 y en
    ies.losada.dieiguez@edu.xunta.es</address>
  </body>
</html>
```

*Puedes contactar con el centro educativo en el teléfono 886 151 924 y en  
ies.losada.dieiguez@edu.xunta.es*



# Etiquetas de bloque: citas/referencias

- Etiqueta `<blockquote>`.
- Representa una cita textual.
- Con el atributo `cite` podemos indicar la fuente del texto citado.

# Etiquetas de bloque: citas/referencias

```
<!DOCTYPE html>

<html>
  <head>
    <meta charset="utf-8"/>
    <title>Citas HTML</title>
  </head>
  <body>
    <p> A continuación se muestra una cita de la
    página web de Debian:</p>
    <blockquote cite=
    "https://www.debian.org/intro/philosophy">El
    proyecto Debian es una asociación de personas
    que comparten un objetivo: queremos crear un
    sistema operativo libre, disponible para todo
    el mundo. Ahora bien, cuando utilizamos el
    término «libre» no estamos hablando de dinero,
    sino que nos referimos a la libertad del
    software</blockquote>
  </body>
</html>
```

A continuación se muestra una cita de la página web de Debian:

El proyecto Debian es una asociación de personas que comparten un objetivo: queremos crear un sistema operativo libre, disponible para todo el mundo. Ahora bien, cuando utilizamos el término «libre» no estamos hablando de dinero, sino que nos referimos a la libertad del software

# Etiquetas de bloque: notas

- Etiqueta `<details>`.
- Nos permite crear notas aclaratorias desplegadas.
- Podemos usar la etiqueta `<summary>` para cambiar el texto predefinido del desplegable.

# Etiquetas de bloque: notas

```
<!DOCTYPE html>

<html>
  <head>
    <meta charset="utf-8"/>
    <title>Detalles HTML</title>
  </head>
  <body>
    <details> La etiqueta details puede no
      funcionar correctamente en algunos
      navegadores...</details>
  </body>
</html>
```

► Detalles



▼ Detalles

La etiqueta details puede no funcionar correctamente en algunos navegadores...

# Etiquetas de bloque: notas

```
<!DOCTYPE html>

<html>
  <head>
    <meta charset="utf-8"/>
    <title>Detalles HTML</title>
  </head>
  <body>
    <details>
      <summary>Despliega para más
      detalles</summary>
      La etiqueta details puede no
      funcionar correctamente en
      algunos navegadores...
    </details>
  </body>
</html>
```

► Despliega para más detalles



▼ Despliega para más detalles

La etiqueta details puede no funcionar correctamente en algunos navegadores...

# Etiquetas de bloque: otras etiquetas

- Etiqueta `<br>`: introduce un salto de línea. No se recomienda usar, ya que disponemos de herramientas en CSS para controlar los saltos de línea sin tener que especificarlos en el documento HTML.
- Etiqueta `<hr>`: línea divisoria horizontal. Semánticamente indica un cambio de tema, generalmente entre párrafos.
- Etiqueta `<pre>`: permite la escritura de forma literal entre su apertura y su cierre.

# Etiquetas de bloque: otras etiquetas

```
<body>
  <span>El texto de esta etiqueta se continúa con el de la siguiente </span>
  <span>porque están dentro de etiquetas de línea, pero puedo usar <lt;br> para que la siguiente etiqueta
  </span> <br>
  <span>cambie de línea</span>
  <p>También puedo usar la etiqueta <lt;hr> para introducir una línea vertical que indica un cambio de tema
  </p> <hr>
  <pre>Por último, puedo usar <lt;pre> para escribir de forma literal</pre>
</body>
```

El texto de esta etiqueta se continúa con el de la siguiente porque están dentro de etiquetas de línea, pero puedo usar `<br>` para que la siguiente etiqueta cambie de línea

También puedo usar la etiqueta `<hr>` para introducir una línea vertical que indica un cambio de tema

---

Por último, puedo usar `<pre>` para escribir de forma literal

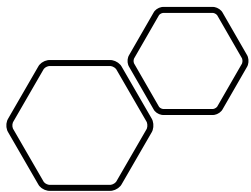
# Etiquetas de bloque: contenedores

- La etiqueta `<div>` define una etiqueta de bloque sin modificación del aspecto, ni contenido semántico más allá de ser un contenedor.
  - Misma función que `<span>`, pero como elemento de bloque.
  - Es útil porque mediante CSS podemos cambiar su representación gráfica e incluso puede añadirse funcionalidad con JavaScript.
  - Muy usado para agrupar elementos cuando no hay una etiqueta semántica que se ajuste a lo que buscamos.
  - En HTML5 tenemos etiquetas equivalentes a `<div>` pero con contenido semántico, que son las que usamos para maquetar las principales secciones de nuestro documento.



# Etiquetas de bloque: contenedores

- Mismo efecto que `<div>`, pero con contenido semántico:
- Etiqueta `<header>`
- Etiqueta `<main>`
- Etiqueta `<nav>`
- Etiqueta `<aside>`
- Etiqueta `<section>`
- Etiqueta `<article>`
- Etiqueta `<footer>`



# Etiquetas de línea



# Etiquetas de línea: contenedores

- La etiqueta `<span>` define una etiqueta de línea sin modificación del aspecto, ni contenido semántico más allá de ser un contenedor.
  - Misma función que `<div>`, pero como elemento de línea.
  - Es útil porque mediante CSS podemos cambiar su representación gráfica e incluso puede añadirse funcionalidad con JavaScript.
  - Opción interesante si tenemos contenido textual “huérfano” y no es semánticamente razonable incluirlo en otro tipo de etiqueta de línea.

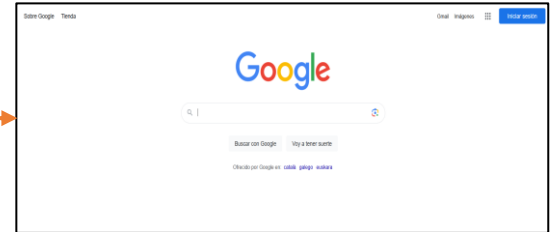
# Etiquetas de línea: enlaces

- Etiqueta `<a>`: representa un enlace a una URL de internet.
  - Al pulsar el enlace el navegador se dirige al destino indicado en el mismo.
  - Usamos el atributo `href` para indicar la URL de destino.
  - También podemos referenciar a un elemento concreto dentro de una URL, ya sea éste otro documento o bien el mismo que contiene el elemento `<a>`, usando `#identificador` después de la URL.
  - El aspecto por defecto de un enlace es color azul y subrayado.

# Etiquetas de línea: enlaces

```
<body>
  <h3>Prueba de enlaces</h3>
  <a id="inicio" href="#fondo">Ir al fondo</a>
  <a href="https://www.google.es/">Ir a Google</a>
  <a href="mienlace.html">Ir a la otra página</a>
  .
  .
  .
  .
  <a id="fondo" href="#inicio">Ir al inicio</a>
</body>
```

www.google.es



mienlace.html

Este es el destino del enlace

# Etiquetas de línea: abreviaturas

- La etiqueta `<abbr>` permite indicar un acrónimo o abreviatura. Se usa el atributo `title` para definir la expresión o nombre no abreviado.

```
<body>
  <p>Segun la <abbr title="Organización Mundial de la Salud">OMS</abbr>
  Los adultos de 18 a 64 años dediquen como mínimo 150 minutos semanales a
  la práctica de actividad física aeróbica, de intensidad moderada, o bien
  75 minutos de actividad física aeróbica vigorosa cada semana, o bien una
  combinación equivalente de actividades moderadas y vigorosas</p>
</body>
```

Segun la OMS Los adultos de 18 a 64 años dediquen como mínimo 150 minutos semanales a la práctica de actividad física aeróbica, de intensidad moderada, o bien 75 minutos de actividad física aeróbica vigorosa cada semana, o bien una combinación equivalente de actividades moderadas y vigorosas

# Etiquetas de línea: texto

- `<strong>` indica texto importante. Visualmente representa el texto en negrita
- `<em>` permite enfatizar un texto. Visualmente aparece en cursiva.
- `<b>` **bold**, el contenido aparecerá en negrita. Igual que strong, pero sin contenido semántico.
- `<i>` *italic*, el contenido aparecerá en cursiva. Igual que em, pero sin contenido semántico.
- `<s>` tacha el texto que contiene.
- `<u>` subraya el texto que contiene.

# Etiquetas de línea: texto

- `<q>` indica texto citado de forma literal. Visualmente aparece entrecomillado.
- `<mark>` resalta un texto, de la misma forma que haríamos con un rotulador para destacar.
- `<small>` indica una anotación. Visualmente se representa mediante un texto de menor tamaño.
- `<sub>` indica que su contenido es un subíndice.
- `<sup>` indica que su contenido es un superíndice.



# Etiquetas de línea: texto

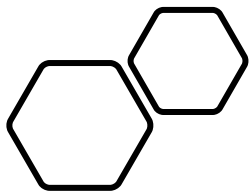
- `<code>` indica que el contenido es código fuente de un lenguaje de programación, y usa una tipografía adecuada, generalmente monoespaciada.
- `<cite>` indica una referencia a una fuente de información
- `<kbd>` hace referencia a una tecla o combinación de éstas.

# Etiquetas de línea: semánticas

- `<data>` facilita el procesamiento automático de datos, vinculando un elemento con una asignación más comprensible por una máquina. Se usa el atributo `data`.
- `<dfn>` indica que el elemento que contiene es un término que vamos a definir. Útil para el procesamiento automático y el posicionamiento de la página.
- `<time>` representa una fecha o un periodo de tiempo. Su uso es puramente semántico.

# Etiquetas de línea: semánticas

- `<del>` indica que un contenido ha sido eliminado o actualizado. Visualmente lo tacha. El atributo `cite` nos permite indicar mediante un enlace el porqué del cambio, mientras que `datetime` contiene la fecha en la que se produjo el cambio
- `<ins>` indica que el elemento que contiene es una nueva inserción (lo contrario que `del`). Visualmente aparece como texto subrayado. También usa los atributos `cite` y `datetime`.



# Tablas en HTML

# Tablas en HTML

- Las etiquetas básicas para crear tablas en HTML son `<table>`, `<tr>`, `<td>`, `<th>`.
- La etiqueta `<table>` define la tabla. Podemos pensar en ella como el recuadro que rodea el conjunto de las celdas de una tabla.
- Dentro de `<table>` iremos declarando cada fila de la tabla con la etiqueta `<tr>`.
- Dentro de cada fila de la tabla, iremos declarando cada celda con `<td>`. Si es la primera fila de la tabla, podemos usar `<th>` en lugar de `<td>` para indicar que es la cabecera de la tabla

# Tablas en HTML

```
<table>
  <tr>
    <td>Celda 1.1</td>
    <td>Celda 1.2</td>
    <td>Celda 1.3</td>
  </tr>
  <tr>
    <td>Celda 2.1</td>
    <td>Celda 2.2</td>
    <td>Celda 2.3</td>
  </tr>
  <tr>
    <td>Celda 3.1</td>
    <td>Celda 3.2</td>
    <td>Celda 3.3</td>
  </tr>
</table>
```

Celda 1.1	Celda 1.2	Celda 1.3
Celda 2.1	Celda 2.2	Celda 2.3
Celda 3.1	Celda 3.2	Celda 3.3

```
<table border="1">
  <tr>
    <td>Celda 1.1</td>
    <td>Celda 1.2</td>
    <td>Celda 1.3</td>
  </tr>
  <tr>
    <td>Celda 2.1</td>
    <td>Celda 2.2</td>
    <td>Celda 2.3</td>
  </tr>
  <tr>
    <td>Celda 3.1</td>
    <td>Celda 3.2</td>
    <td>Celda 3.3</td>
  </tr>
</table>
```

Celda 1.1	Celda 1.2	Celda 1.3
Celda 2.1	Celda 2.2	Celda 2.3
Celda 3.1	Celda 3.2	Celda 3.3

# Tablas en HTML

- En versiones anteriores a HTML5 no era infrecuente usar tablas para delimitar las diferentes secciones de una página web.
- Con la llegada de HTML5 esta práctica ha quedado obsoleta...
- ...salvo en las *newsletters*, que muchas veces son visualizadas en gestores de correo electrónico basadas en navegadores antiguos, sin soporte de HTML5.

# Tablas en HTML

- Además de las etiquetas básicas, existen otras etiquetas para la creación de tablas en HTML:
- `<caption>`: es el título descriptivo de la tabla.
  - Se debe declarar al principio de ésta, justo después de la etiqueta de apertura de `<table>`.
  - Por defecto se muestra en la parte superior de la tabla.
- `<colgroup>`: permite agrupar una o varias columnas (mediante el atributo `span`) para modificar atributos de una o varias columnas, de izquierda a derecha. Pueden funcionar por si mismas o contener etiquetas `<col>`.
- `<col>`: representa una columna. Debe estar contenida en un elemento `<colgroup>` y permite modificar atributos de una columna.



# Tablas en HTML

```
<table border="1">
<colgroup style="width:150px;">
  <col style="background-color:cyan;"></col>
  <col style="background-color:orange;"></col>
  <col style="background-color:lime;"></col>
</colgroup>
<tr>
  <td>Celda 1.1</td>
  <td>Celda 1.2</td>
  <td>Celda 1.3</td>
</tr>
<tr>
  <td>Celda 2.1</td>
  <td>Celda 2.2</td>
  <td>Celda 2.3</td>
</tr>
<tr>
  <td>Celda 3.1</td>
  <td>Celda 3.2</td>
  <td>Celda 3.3</td>
</tr>
</table>
```

Celda 1.1	Celda 1.2	Celda 1.3
Celda 2.1	Celda 2.2	Celda 2.3
Celda 3.1	Celda 3.2	Celda 3.3

# Tablas en HTML

- `<thead>`: la cabecera de la tabla. Debería contener solo la primera fila de la tabla. Se declara, una única vez por tabla, antes de `<tbody>`.
- `<tbody>`: el cuerpo de una tabla, que contiene las celdas con los datos. Puede haber más de un `<tbody>` por tabla, por ejemplo, para separar diferentes secciones.
- `<tfoot>`: el pie de tabla. Solo uno por tabla. Representa la última fila, y suele usarse para indicar información resumida o como el total del resto de filas de la tabla.

# Tablas en HTML

- Existen 2 atributos importantes que podemos aplicar a `<th>` y `<td>`:
  - **colspan**: número de celdas en horizontal, a la derecha, que vamos a agrupar.
  - **rowspan**: número de celdas en vertical, hacia abajo, que vamos a agrupar.
- Por defecto, tanto colspan como rowspan son 1.

```
<table border="1">
  <tr>
    <td colspan="1" rowspan="1">Celda 1.1</td>
    <td colspan="1" rowspan="1">Celda 1.2</td>
    <td colspan="1" rowspan="1">Celda 1.3</td>
  </tr>
  <tr>
    <td colspan="1" rowspan="1">Celda 2.1</td>
    <td colspan="1" rowspan="1">Celda 2.2</td>
    <td colspan="1" rowspan="1">Celda 2.3</td>
  </tr>
  <tr>
    <td colspan="1" rowspan="1">Celda 3.1</td>
    <td colspan="1" rowspan="1">Celda 3.2</td>
    <td colspan="1" rowspan="1">Celda 3.3</td>
  </tr>
</table>
```

=

```
<table border="1">
  <tr>
    <td>Celda 1.1</td>
    <td>Celda 1.2</td>
    <td>Celda 1.3</td>
  </tr>
  <tr>
    <td>Celda 2.1</td>
    <td>Celda 2.2</td>
    <td>Celda 2.3</td>
  </tr>
  <tr>
    <td>Celda 3.1</td>
    <td>Celda 3.2</td>
    <td>Celda 3.3</td>
  </tr>
</table>
```

Celda 1.1	Celda 1.2	Celda 1.3
Celda 2.1	Celda 2.2	Celda 2.3
Celda 3.1	Celda 3.2	Celda 3.3

# Tablas en HTML. Atributo **colspan**

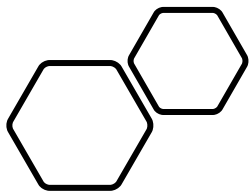
```
<table border="1">
  <tr>
    <td colspan="2">Celda 1.1</td>
    <td colspan="1">Celda 1.2</td>
    <td colspan="1">Celda 1.3</td>
  </tr>
  <tr>
    <td colspan="1">Celda 2.1</td>
    <td colspan="2">Celda 2.2</td>
    <td colspan="1">Celda 2.3</td>
  </tr>
  <tr>
    <td colspan="1">Celda 3.1</td>
    <td colspan="1">Celda 3.2</td>
    <td colspan="2">Celda 3.3</td>
  </tr>
</table>
```

Celda 1.1	Celda 1.2	Celda 1.3
Celda 2.1	Celda 2.2	Celda 2.3
Celda 3.1	Celda 3.2	Celda 3.3

# Tablas en HTML. Atributo **rowspan**

```
<table border="1">
  <tr>
    <td rowspan="2">Celda 1.1</td>
    <td rowspan="1">Celda 1.2</td>
    <td rowspan="1">Celda 1.3</td>
  </tr>
  <tr>
    <td rowspan="1">Celda 2.1</td>
    <td rowspan="2">Celda 2.2</td>
    <td rowspan="1">Celda 2.3</td>
  </tr>
  <tr>
    <td rowspan="1">Celda 3.1</td>
    <td rowspan="1">Celda 3.2</td>
    <td rowspan="2">Celda 3.3</td>
  </tr>
</table>
```

Celda 1.1	Celda 1.2	Celda 1.3	
	Celda 2.1		Celda 2.3
Celda 3.1	Celda 3.2	Celda 2.2	Celda 3.3



# Formularios HTML

# Formularios HTML: elementos básicos

- Uno de los principales usos de HTML es la recogida de información.
- Un ejemplo de formulario es cuando un usuario quiere crear una cuenta en una página web, para lo que requerimos ciertos datos personales.
- Un ejemplo menos intuitivo de formulario es cuando necesitamos hacer una búsqueda por un término o términos indicados por el usuario.
- En todo caso, la etiqueta que empleamos para definir un formulario es `<form>`.

# Formularios HTML: elementos básicos

- La etiqueta `<form>` debe rodear a todos los elementos del formulario.
- La etiqueta `<form>` tiene dos atributos importantes: `action` y `method`.
- Con `action` indicamos la página web que recibirá los datos del formulario una vez enviado.
- Con `method` indicamos el método con el que enviamos la información. Existen dos métodos: `get` y `post`.
- Deberíamos usar POST cuando solicitamos información del usuario.
- Deberíamos usar GET cuando la información introducida no sea comprometida: en los campos de búsqueda.



# Formularios HTML: elementos básicos

- Una vez declarado el formulario con `<form>`, necesitamos introducir elementos que nos permitan recabar la información necesaria.
- Esto se realiza con etiquetas HTML en las que es posible introducir estos datos.
- Muchos de estos elementos se forman con la etiqueta `<input>` (entrada), seguido del atributo `type`.
- Los elementos que se van a enviar al destino deben tener definido el atributo `name`.

# Formularios HTML: elementos básicos

- Campos de texto: `<input type="text">`
  - Declara un campo de texto de una línea. Suele ser suficiente para la mayoría de los datos del usuario. Con CSS podemos cambiar su longitud.
- Contraseñas: `<input type="password">`
  - Se diferencia del anterior en que oculta el texto que estamos escribiendo.
- Botones: `<input type="button">`
  - Declara un botón, cuya acción podemos configurar con JavaScript.
  - Equivalente a usar la etiqueta `<button></button>`

# Formularios HTML: elementos básicos

- Botón de reset: `<input type="reset">`
  - Declara un botón, que al pulsarse deja los elementos del formulario en su valor predefinido.
- Botón de envío: `<input type="submit">`
  - Envía el formulario a la página indicada en el atributo `action` de `<form>`.
- Información oculta: `<input type="hidden">`
  - El usuario no verá este elemento. Suele usarse para enviar al servidor información recogida de forma automática, sin la intervención del usuario.

# Formularios HTML: elementos básicos

```
<body>
  <form action="reception.html" method="post" enctype="multipart/form-data">
    <p>Nombre:</p>
    <input type="text" placeholder="Nombre" name="nombre"/>
    <p>Contraseña:</p>
    <input type="password" placeholder="Contraseña" name="contrasenha"/>
    <!-- El siguiente botón no hace nada, pues no tenemos ningún script -->
    <input type="button" value="No funciona"/>
    <input type="reset" value="Borrar"/>
    <input type="submit" value="Enviar formulario"/>
    <input type="hidden" name="oculto" value="Todo lo que pongo aquí no es visible"/>
  </form>
</body>
```

Nombre:

Contraseña:

# Formularios HTML: elementos básicos

- Imágenes: `<input type="image">`
  - Nos permite poner una imagen, que en el contexto del formulario actuará como el botón de envío.
- Archivos: `<input type="file">`
  - Permite al usuario indicar un archivo que enviar con el formulario. No es compatible con el método GET.

```
<body>
  <form action="reception.html" method="post" enctype="multipart/form-data">
    <input type="file" name="adjunto"/>
    <input type="image" height="25px" src="img/enviar.png"/>
  </form>
</body>
```

Seleccionar archivo Ninguno archivo selec.

# Formularios HTML: elementos básicos

- Selección múltiple: `<input type="checkbox">`
  - Nos permite poner una caja que podemos marcar o desmarcar. Como cada *checkbox* es individual podemos marcar una, varias o ninguna.
- Selección excluyente: `<input type="radiobutton">`
  - Permite crear opciones excluyentes entre sí: tiene que marcarse una opción siempre, y no puede marcarse más de una.
  - Para que funcione correctamente, las opciones que forma el grupo tienen que tener el mismo atributo `name`;

# Formularios HTML: elementos básicos

```
<body>
  <form action="reception.html" method="post" enctype="multipart/form-data">
    <h3>Puestos de trabajo solicitados:</h3>
    <input type="checkbox" name="directivo" value="direccion"/><span>Dirección</span>
    <input type="checkbox" name="tecnico" value="tecnico"/><span>Técnico</span>
    <input type="checkbox" name="empleado" value="empleado"/><span>Empleado</span>
    <h3>Género </h3>
    <input type="radio" name="genero" value="undefined" checked="checked"/> <span>Prefiero no
indicarlo</span><br/>
    <input type="radio" name="genero" value="hombre"/> <span>Hombre</span><br/>
    <input type="radio" name="genero" value="mujer"/> <span>Mujer</span><br/>
  </form>
</body>
```

**Puestos de trabajo solicitados:**

☐ Dirección ☐ Técnico ☐ Empleado

**Género**

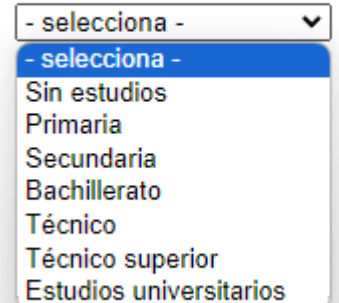
☒ Prefiero no indicarlo  
☐ Hombre  
☐ Mujer

# Formularios HTML: elementos básicos

- Con la etiqueta `<select>` podemos crear una lista desplegable.
  - Cada elemento de la lista se indica con la etiqueta `<option>`

```
<body>
  <form action="reception.html" method="post" enctype="multipart/form-data">
    <h3>Nivel de estudios:</h3>
    <select name="estudios">
      <option value="" selected="selected">- selecciona -</option>
      <option value="sin_estudios">Sin estudios</option>
      <option value="primaria">Primaria</option>
      <option value="secundaria">Secundaria</option>
      <option value="bachillerato">Bachillerato</option>
      <option value="tecnico">Técnico</option>
      <option value="tecnico_superior">Técnico superior</option>
      <option value="universitarios">Estudios universitarios</option>
    </select>
  </form>
</body>
```

Nivel de estudios:





# Formularios HTML: elementos básicos

- Con la etiqueta `<textarea>` podemos definir un campo de texto con más espacio para escribir que `<input type="text">`

```
<body>
  <form action="reception.html" method="post" enctype="multipart/form-data">
    <h3>Input de tipo texto:</h3>
    <input type="text" name="nombre"/>
    <h3>Área de texto:</h3>
    <textarea name="presentacion" cols="40" rows="5"></textarea>
  </form>
</body>
```

Input de tipo texto:

Área de texto:

# Formularios HTML: elementos básicos

- Atributos interesantes en etiquetas de formularios:
  - **name**: debe estar presente si queremos que la información de la etiqueta se envíe al destino.
  - **placeholder**: nos permite poner un texto de ejemplo.
  - **required**: indica que el elemento es obligatorio, no se puede enviar el formulario sin introducir información en él.
  - **autocomplete**: indica si los campos se pueden autocompletar con los introducidos en ocasiones anteriores.
  - **value**: es el valor por defecto del componente.

# Formularios HTML: otros elementos

- Con respecto a `<input>`, tenemos algunas opciones más en función del atributo `type`, algunas de las cuales llegaron con la versión 5 de HTML. Muchas de ellas incorporan restricciones para controlar que el tipo de entrada se ajuste a un determinado formato o criterio.
  - `<input type="email">`: comprueba que tenga formato de email.
  - `<input type="tel">`: comprueba que tenga formato de teléfono.
  - `<input type="url">`: comprueba que tenga formato de URL.
  - `<input type="number">`: comprueba que sea un número, controla entrada.
  - `<input type="range">`: establece un *slider* entre valores máximo y mínimo.
  - `<input type="search">`: campo para realizar una búsqueda (similar a text).
  - `<input type="color">`: permite seleccionar de forma interactiva un color.

# Formularios HTML: otros elementos

- `<input type="date">`: fecha.
- `<input type="time">`: hora.
- `<input type="datetime-local">`: fecha y hora.
- `<input type="week">`: año y número de semana dentro de ese año.
- `<input type="month">`: mes y año.

# Formularios HTML: otros elementos

- Todas las etiquetas `<input>` tienen algunos atributos comunes que nos puede interesar aplicar:
  - **autofocus**: pone el foco automáticamente en el elemento que incorpora este atributo.
  - **disabled**: desactiva el componente que lo incorpore.
  - **readonly**: si incorporamos este atributo, el contenido del atributo **value** será visible pero no editable.
  - **form**: contiene una referencia al id del formulario.
  - **list**: no disponible para todos los tipos de `<input>`. Permite enlazar el componente con una lista de opciones definida en una etiqueta `<datalist>`

# Formularios HTML: otros elementos

- Con `<datalist>` podemos definir una lista de elementos (cada uno contenido en una etiqueta `<option>`), que posteriormente podemos usar en otras etiquetas `<input>` del documento.

```
<body>
  <datalist id="modulos">
    <option>Programación</option>
    <option>Bases de datos</option>
    <option>Linguaxes de marcas</option>
    <option>Contornos de desenvolvemento</option>
    <option>Sistemas informáticos</option>
    <option>FOL</option>
  </datalist>
  <form action="reception.html" method="post" enctype="multipart/form-data">
    <h3>Este input de tipo texto es como los que hemos visto hasta ahora:</h3>
    <input type="text" name="nombre"/>
    <h3>Este input de tipo texto tiene otro comportamiento:</h3>
    <input type="text" name="asignaturas" list="modulos"/>
  </form>
</body>
```

Este input de tipo texto es como los que hemos visto hasta ahora:

Este input de tipo texto tiene otro comportamiento:

Programación

Bases de datos

Linguaxes de marcas

Contornos de desenvolvemento

Sistemas informáticos

FOL

# Formularios HTML: otros elementos

- La etiqueta `<fieldset>` permite agrupar varios elementos de un formulario
- Si usamos la etiqueta `<legend>` dentro de `<fieldset>` podemos establecer un nombre para el grupo, que figurará en la parte superior.

# Formularios HTML: otros elementos

```
<form action="reception.html" method="get">
  <fieldset>
    <legend>Login</legend>
    <h4>Nombre de usuario:</h4>
    <input type="text" name="nombre"/>
    <h4>Contraseña:</h4>
    <input type="password" name="contrasinal"/>
  </fieldset>
  <fieldset>
    <legend>Datos de contacto</legend>
    <h4>Correo electrónico:</h4>
    <input type="email" name="ecorreos"/>
    <h4>Número de teléfono:</h4>
    <input type="tel" name="telefono"/>
  </fieldset>
</form>
```

Login

Nombre de usuario:

Contraseña:

Datos de contacto

Correo electrónico:

Número de teléfono:



# Formularios HTML: otros elementos

- `<optgroup>` se puede usar dentro de `<select>` para establecer categorías o grupos de elementos `<option>`. Se usa el atributo **label** para indicar el nombre del grupo.

```
<body>
  <form action="reception.html" method="post" enctype="multipart/form-data">
    <h3>Nivel de estudios:</h3>
    <select name="estudios">
      <option value="" selected="selected">- selecciona -</option>
      <optgroup label="Enseñanza obligatoria">
        <option value="sin_estudios">Sin estudios</option>
        <option value="primaria">Primaria</option>
        <option value="secundaria">Secundaria</option>
      </optgroup>
      <optgroup label="Enseñanza postobligatoria">
        <option value="bachillerato">Bachillerato</option>
        <option value="tecnico">Técnico</option>
        <option value="tecnico_superior">Técnico superior</option>
        <option value="universitarios">Estudios universitarios</option>
      </optgroup>
    </select>
  </form>
</body>
```

Nivel de estudios:

- selecciona - ▼

- selecciona -

**Enseñanza obligatoria**

Sin estudios

Primaria

Secundaria

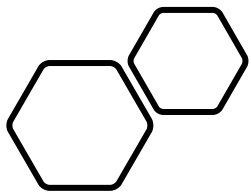
**Enseñanza postobligatoria**

Bachillerato

Técnico

Técnico superior

Estudios universitarios



# Contenidos incrustados en HTML

# Contenido incrustado

- El contenido incrustado se refiere a un contenido que está alojado en una ubicación diferente a la del propio documento HTML (es decir, es externo).
- Como no está en el propio documento, podemos modificar la fuente original y el contenido incrustado reflejará dichos cambios.
- El contenido incrustado puede ser una imagen, un vídeo, un fragmento de HTML, etc.
- Atributos destacables: **src** (fuente), **type** (tipo de contenido) y **width** y **height** (determinan el ancho y alto del contenido).

# Contenido incrustado

- La etiqueta `<iframe>` nos permite incrustar un documento, o parte de él, en nuestra página.
- Atributos destacables: `allowfullscreen`.

# Contenido incrustado

```
<body>  
  <h3>El siguiente contenido es externo a mi documento html:  
</h3>  
  <iframe width="560" height="315" src=  
    "https://www.youtube.com/embed/WTko1H0IRCY" frameborder="0"  
    allow="accelerometer; autoplay; clipboard-write;  
    encrypted-media; gyroscope; picture-in-picture"  
    allowfullscreen></iframe>  
</body>
```

El siguiente contenido es externo a mi documento html:



# Contenido incrustado

```
<body>
  <h3>El siguiente contenido es externo a mi documento html:</h3>
  <iframe src=
    "https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d2932.508111
    7587783!2d-8.507802924331388!3d42.6929608140056!2m3!1f0!2f0!3f0!3m
    2!1i1024!2i768!4f13.1!3m3!1m2!1s0xd2f081beaaaaaab%3A0xf7870d702dd9
    4130!2s%5BIES%5D%20Instituto%20de%20Educaci%C3%B3n%20Secundaria%20
    Ant%C3%B3n%20Losada%20Di%C3%A9go%20Guez!5e0!3m2!1ses!2ses!4v16965036923
    28!5m2!1ses!2ses" width="400" height="300" style="border:0;"
    allowfullscreen="" loading="lazy" referrerpolicy=
    "no-referrer-when-downgrade"></iframe>
</body>
```

El siguiente contenido es externo a mi documento html:



# Contenido incrustado

- La etiqueta `<embed>` nos permite crear un área para acoplar un *plug-in* generado por otra aplicación, como, por ejemplo, un *applet* de Java.
- Puede emplearse también para incrustar un vídeo o imagen, tanto procedente de nuestro servidor como de uno externo.

## Warning

Most browsers no longer support Java Applets and Plug-ins.

ActiveX controls are no longer supported in any browsers.

The support for Shockwave Flash has also been turned off in modern browsers.

## Suggestion

To display a picture, it is better to use the `<img>` tag.

To display HTML, it is better to use the `<iframe>` tag.

To display video or audio, it is better to use the `<video>` and `<audio>` tags.

[https://www.w3schools.com/tags/tag\\_embed.asp](https://www.w3schools.com/tags/tag_embed.asp)

# Contenido incrustado

- La etiqueta `<object>` nos permite insertar un objeto externo (imagen, vídeo, código HTML o un *plug-in*). La etiqueta `<param>` permite definir los parámetros del contenido incrustado.

## Warning

Most browsers no longer support Java Applets and Plug-ins.

ActiveX controls are no longer supported in any browsers.

The support for Shockwave Flash has also been turned off in modern browsers.

## Suggestion

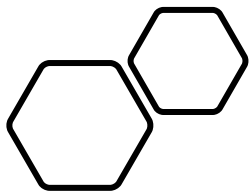
To display a picture, it is better to use the `<img>` tag.

To display HTML, it is better to use the `<iframe>` tag.

To display video or audio, it is better to use the `<video>` and `<audio>` tags.

[https://www.w3schools.com/tags/tag\\_object.asp](https://www.w3schools.com/tags/tag_object.asp)





# Multimedia en HTML

# Multimedia

- Los elementos multimedia nos permiten añadir imágenes, audio y vídeo a nuestros documentos HTML. Las imágenes han estado presentes desde las primeras versiones de HTML, mientras que la etiquetas para audio y vídeo fueron añadidas en HTML5.
- Anteriormente el vídeo y audio se añadían como elementos incrustados mediante etiquetas como `<embed>` o `<object>`.

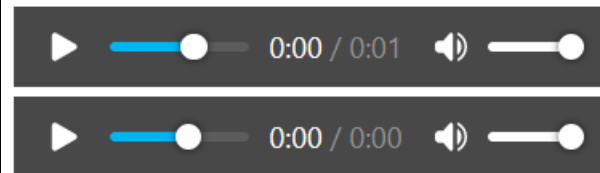
# Multimedia: audio

- Etiqueta `<audio>`: permite insertar recursos sonoros en los documentos HTML.
- Permite asignar el audio mediante el atributo `src` o la etiqueta `<source>`.
- Atributos destacables:
  - `src`: permite establecer la ruta al recurso de audio.
  - `autoplay`: indica que el contenido se reproduzca de forma automática al cargar la página. Es una mala decisión de diseño activar esta opción y muchos navegadores la tienen bloqueada.
  - `loop`: indica que el audio se va a reproducir en bucle.
  - `muted`: silencia el audio por defecto.
  - `controls`: muestra los controles para gestionar la reproducción del audio.

# Multimedia: audio

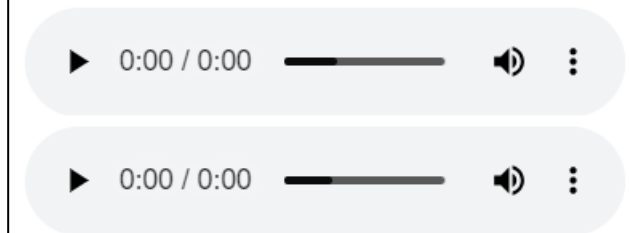
```
<body>
  <h3>Tenemos dos formas de añadir audio.</h3>
  <audio src="bounce.mp3" type="audio/mpeg" autoplay muted controls loop>
    Aquí puedo indicar un mensaje, para el caso de que el navegador no
    pueda reproducir el audio</audio>
  <audio>
    <source src="bounce.mp3" type="audio/mpeg">
    <source src="bounce.ogg" type="audio/ogg">
    Aquí puedo indicar un mensaje, para el caso de que el navegador no
    pueda reproducir el audio
  </audio>
</body>
```

Tenemos dos formas de añadir audio.



Firefox

Tenemos dos formas de añadir audio.



Google Chrome, Edge y Opera

# Multimedia: vídeo

- Etiqueta `<video>`: permite insertar recursos de vídeo en los documentos HTML.
- Permite asignar el vídeo mediante el atributo `src` o la etiqueta `<source>`.

# Multimedia: vídeo

- Atributos destacables:
  - **src**: permite establecer la ruta al vídeo.
  - **autoplay**: indica que el contenido se reproduzca de forma automática al cargar la página. Es una mala decisión de diseño activar esta opción y muchos navegadores la tienen bloqueada.
  - **loop**: indica que el audio se va a reproducir en bucle.
  - **muted**: silencia el audio por defecto.
  - **controls**: muestra los controles para gestionar la reproducción del vídeo.
  - **poster**: si se le indica la ruta de una imagen la establece como fotograma por defecto del vídeo mientras no se empieza a reproducir.
  - **width** y **height**: permiten ajustar la anchura y altura del vídeo.

# Multimedia: vídeo

```
<body>
  <h3>Tenemos dos formas de añadir video.</h3>
  <video src="multimedia/homer.mp4" autoplay controls loop>Aquí puedo
  indicar un mensaje, para el caso de que el navegador no pueda
  reproducir el video</video>
  <video poster="multimedia/miniatura.png" controls loop>
    <source src="multimedia/homer.mp4" type="video/mp4"/>
    <source src="multimedia/homer.webm" type="video/webm"/>
  </video>
</body>
```

Tenemos dos formas de añadir video.



# Multimedia: imagen

- Etiqueta `<img>`: permite insertar una imagen en el documento HTML indicando su ruta con el atributo `src`.
- Atributos destacables:
  - `alt`: un texto descriptivo de la imagen, para el caso de que la imagen no pueda ser percibida. Además, ayuda a potenciar la visibilidad de nuestra página en buscadores.
  - `width` y `height`: permiten ajustar la anchura y altura de la imagen.
  - `usemap`: permite asignar un mapa de áreas para usar como enlaces.
- Etiquetas relacionadas: `<map>` y `<area>`.



# Multimedia: imagen

```
<body>
  <h3>La siguiente imagen tiene enlaces asociados a áreas</h3>
  <map id="mapaEnlaces">
    <area href="enlace1.html" alt="Enlace 1" shape="rect" coords=
      "10,10,290,48"/>
    <area href="enlace2.html" alt="Enlace 2" shape="rect" coords=
      "10,77,290,116"/>
  </map>
  
</body>
```

La siguiente imagen tiene enlaces asociados a áreas

Primer enlace

Segundo enlace

# Multimedia: otras etiquetas

- La etiqueta `<svg>` permite describir trazos vectoriales usando este formato. El formato SVG lo veremos en la última UD del curso.
- La etiqueta `<canvas>` nos permite definir un lienzo en el que podemos mostrar gráficos y animaciones mediante JavaScript.
- Iconos de favoritos: podemos asignar un icono para que aparezca como miniatura en la ventana o pestaña correspondiente del navegador (favicon). Esto se realiza usando la etiqueta `<link>` dentro de `<head>`:

```
<link rel="icon" href="ruta-icono.extension"/>
```