

UT 2.2 - Crear interfaces web en HTML



Indice

1. Conceptos básicos
 2. !DOCTYPE y metaelementos en HTML5
 3. Secciones
 4. Agrupar contenido
 5. Semántica a nivel de texto
 6. Editar contenido
 7. Contenido embebido
 8. Enlaces
 9. Tablas
 10. Formularios
- Referencias

1. Conceptos básicos

Versiones

HTML 4.01

- Última versión completa de HTML 4 (1999)

XHTML 1.0 y 1.1

- *Extensible HTML*.
- Variante de HTML. Más estricto que HTML. Si una página está escrita en XHTML se puede decir también que está escrita en HTML.
- Versiones:
 - XHTML 1.0 y XHTML 1.1
 - XHTML 2.0: Muy estricto, ha sido abandonado.

HTML5 (*sin espacio entre HTML y 5*)

- Recomendación del W3C (28 de octubre de 2014).
- Compatible con XHTML, pero menos estricto que XHTML.

HTML 5.1 2ª ed. (*con espacio entre HTML y 5*)

- Recomendación del W3C (3 de octubre de 2017).

HTML 5.2

- Recomendación del W3C (14 de diciembre de 2017).

1. Conceptos básicos

Version Actual

HTML5 VS HTML Living Standard

La principal diferencia entre HTML5 y el HTML Living Standard radica en cómo se actualizan y mantienen:

HTML5: Es una versión específica. HTML5 introdujo muchas mejoras y nuevas características en comparación con versiones anteriores, como nuevos elementos semánticos, soporte multimedia nativo, y APIs avanzadas¹.

HTML Living Standard: Es una versión en constante evolución del HTML mantenida por el WHATWG (Web Hypertext Application Technology Working Group). A diferencia de HTML5, que es una versión estática, el HTML Living Standard se actualiza continuamente para incorporar nuevas características, mejoras y correcciones de errores.

1. Conceptos básicos

HTML Living Standard, contenidos de: <https://html.spec.whatwg.org/>

- **Infraestructura común:** Terminología y conceptos básicos.
- **Semántica y estructura:** Elementos y atributos de HTML.
- **Microdatos:** Cómo usar microdatos para anotar contenido.
- **Interacción del usuario:** Eventos y manejo de interacciones.
- **Carga de páginas web:** Procesos y técnicas de carga.
- **APIs de aplicaciones web:** APIs disponibles para desarrolladores.
- **Comunicación:** Métodos de comunicación entre documentos y servidores.
- **Trabajadores web:** Uso de Web Workers para tareas en segundo plano.
- **Almacenamiento web:** Técnicas de almacenamiento local y de sesión.
- **Sintaxis de HTML:** Reglas y convenciones de sintaxis.
- **Sintaxis de XML:** Uso de XML en documentos HTML.
- **Renderizado:** Cómo los navegadores renderizan HTML.
- **Características obsoletas:** Elementos y atributos en desuso.
- **Consideraciones de IANA:** Registro de tipos de medios y otros valores.

1. Conceptos básicos

HTML Living Standard <https://html.spec.whatwg.org/>

- **Infraestructura común:** Esta sección incluye definiciones de términos clave, como elementos, atributos y nodos, así como una explicación de cómo se estructuran y manipulan los documentos HTML. También aborda temas como el modelo de objetos del documento (DOM) y la serialización de HTML.
- **Semántica y estructura:** Aquí se detallan los elementos y atributos de HTML que se utilizan para definir la estructura y el significado del contenido de una página web. La semántica es crucial para la accesibilidad y la optimización en motores de búsqueda (SEO). Esta sección explica cómo usar correctamente elementos como encabezados, párrafos, listas, enlaces y tablas para crear documentos bien estructurados y comprensibles.
- **Microdatos:** Los microdatos son una forma de anotar el contenido de HTML con información adicional que puede ser utilizada por motores de búsqueda y otras aplicaciones. Esta sección describe cómo usar los atributos itemscope, itemtype y itemprop para agregar metadatos a los elementos HTML. Los microdatos ayudan a mejorar la visibilidad y la relevancia del contenido en los resultados de búsqueda.

1. Conceptos básicos

HTML Living Standard <https://html.spec.whatwg.org/>

- **Interacción del usuario:** Esta parte del documento se centra en los eventos y el manejo de interacciones del usuario en una página web. Explica cómo los desarrolladores pueden usar eventos como clics, desplazamientos y entradas de teclado para crear experiencias interactivas. También aborda la gestión de eventos y la propagación de eventos, proporcionando una base sólida para el desarrollo de interfaces de usuario dinámicas.
- **Carga de páginas web:** La carga eficiente de páginas web es esencial para una buena experiencia de usuario. Esta sección cubre los procesos y técnicas para optimizar la carga de páginas, incluyendo la gestión de recursos, el uso de caché y la minimización de solicitudes HTTP. También se discuten estrategias como la carga diferida y la carga condicional para mejorar el rendimiento.
- **APIs de aplicaciones web:** HTML proporciona varias APIs que los desarrolladores pueden usar para crear aplicaciones web ricas y funcionales. Esta sección describe APIs como la API de Canvas para gráficos, la API de Geolocalización para obtener la ubicación del usuario y la API de Historial para gestionar el historial de navegación. Estas APIs amplían las capacidades de HTML y permiten la creación de aplicaciones web avanzadas.

1. Conceptos básicos

HTML Living Standard <https://html.spec.whatwg.org/>

- **Comunicación:** La comunicación entre documentos y servidores es fundamental para las aplicaciones web modernas. Esta sección cubre métodos como XMLHttpRequest y Fetch para realizar solicitudes HTTP, así como técnicas para la comunicación entre ventanas y marcos mediante postMessage. También se discuten los WebSockets para la comunicación en tiempo real y las técnicas de CORS para manejar solicitudes entre orígenes.
- **Trabajadores web:** Los Web Workers permiten ejecutar scripts en segundo plano sin bloquear la interfaz de usuario. Esta sección explica cómo crear y usar Web Workers para realizar tareas intensivas en computación, como el procesamiento de datos y la manipulación de imágenes, de manera eficiente. Los Web Workers mejoran el rendimiento y la capacidad de respuesta de las aplicaciones web.
- **Almacenamiento web:** HTML proporciona varias técnicas para almacenar datos en el navegador del usuario. Esta sección cubre el almacenamiento local y de sesión, que permiten guardar datos de manera persistente o temporal. También se discuten las APIs de IndexedDB y Web SQL para el almacenamiento de datos estructurados. Estas técnicas son esenciales para aplicaciones web que necesitan almacenar grandes cantidades de datos o funcionar sin conexión.

1. Conceptos básicos

HTML Living Standard <https://html.spec.whatwg.org/>

- **Sintaxis de HTML:** La sintaxis de HTML define las reglas y convenciones para escribir documentos HTML válidos. Esta sección describe la estructura básica de un documento HTML, incluyendo el uso de etiquetas de apertura y cierre, atributos y entidades de caracteres. También se abordan las diferencias entre HTML y XHTML, y se proporcionan ejemplos de buenas prácticas para escribir código HTML limpio y mantenible.
- **Sintaxis de XML:** Aunque HTML5 no requiere el uso de XML, es posible escribir documentos HTML en una sintaxis compatible con XML. Esta sección explica cómo usar la sintaxis de XML en documentos HTML, incluyendo el uso de declaraciones XML, espacios de nombres y entidades de caracteres. La compatibilidad con XML permite una mayor interoperabilidad con otras tecnologías basadas en XML.
- **Renderizado:** El renderizado es el proceso mediante el cual los navegadores interpretan y muestran el contenido de un documento HTML. Esta sección describe cómo los navegadores construyen el árbol de renderizado a partir del DOM y aplican estilos CSS para presentar el contenido visualmente. También se discuten temas como el flujo de diseño, el reflujo y el repintado, que afectan el rendimiento del renderizado.

1. Conceptos básicos

HTML Living Standard <https://html.spec.whatwg.org/>

- **Características obsoletas:** Con el tiempo, algunos elementos y atributos de HTML se vuelven obsoletos y ya no se recomiendan para su uso. Esta sección lista las características obsoletas y proporciona alternativas modernas. El uso de características actualizadas mejora la compatibilidad y la accesibilidad de los documentos HTML, asegurando que sigan siendo relevantes y funcionales en el futuro.
- **Consideraciones de IANA:** La Internet Assigned Numbers Authority (IANA) gestiona varios registros relacionados con HTML, como los tipos de medios y los valores de atributos. Esta sección describe las consideraciones y procedimientos para registrar nuevos tipos de medios y otros valores en los registros de IANA. La coordinación con IANA asegura que los estándares de HTML se mantengan consistentes y ampliamente reconocidos.

Estructura de un Documento HTML

Un documento HTML se organiza en una jerarquía de elementos anidados.

Declaración del tipo de documento (DOCTYPE): Indica al navegador que se está utilizando HTML5.

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

Elemento <head>: Contiene metadatos sobre el documento, como el título, enlaces a hojas de estilo y scripts.

```
<head>  
  <meta charset="UTF-8">  
  <title>Mi Página Web</title>  
  <link rel="stylesheet" href="styles.css">  
</head>
```

Elemento <body>: La parte visible de la página, como párrafos, imágenes, enlaces, etc.

HTML

```
<body>  
  <h1>Bienvenido a Mi Página Web</h1>  
  <p>Este es un párrafo de ejemplo.</p>  
</body>
```

Etiquetas

- Es: una palabra clave encerrada entre corchetes angulares `<html>`
- Tipos: etiqueta inicial `<html>` y etiqueta final `</html>`.
 - La etiqueta inicial puede tener **atributos**.
`<html lang="es">`
- Etiquetas: sintaxis HTML5:
 - Se pueden escribir en mayúsculas o minúsculas `<html>` o `<HTML>`
 - Se recomienda: **minúsculas** `<html>` (por compatibilidad con versiones anteriores de HTML)

Elementos

- Es: conjunto formado por la etiqueta inicial, la etiqueta final y lo que haya entre ambas.

- Ej. `<head>...</head>`

■ Elementos vacíos

- Elementos que sólo tienen la etiqueta inicial, no tienen etiqueta final.

- Ej. `
`

- Elementos vacíos: sintaxis HTML5: dos formas admitidas:

- `
`
 - `
`

Elegir las etiquetas semánticamente

- Es importante **elegir las etiquetas semánticamente**, es decir, por su significado, no por cómo se va a visualizar (o se suele visualizar) en el navegador.
 - Ej. Las etiquetas <h1>...<h6> se deben utilizar para texto que es cabecera de un determinado nivel, no por el aspecto que le da por defecto el navegador (este se cambia con CSS).
- Un documento con etiquetas semánticamente correctas
 - Permite que los lectores no-humanos, como los **robots de búsqueda**, indexen correctamente el contenido.
 - Se visualiza correctamente en **más navegadores**.

Elementos anteriores a HTML5

■ Elementos obsoletos

- Ya no forman parte del lenguaje HTML5.
- Ej. `<center>`, `<frame>`, `<frameset>`, `<noframes>`, ``

■ Elementos que tienen un significado diferente

- Ej. `<i>` o ``

■ Elementos nuevos

- Ej. `<mark>`, `<address>`, `<time>`

Atributos

- Proporcionan información acerca de un elemento HTML.
 - Ej. Indicar el lenguaje de la página: `<html lang="es">`
- Normalmente tienen la forma
 - **nombre=valor** (ej. `lang="es"`)
- **Atributos: sintaxis HTML5**
 - **Nombre** del atributo: **mayúsculas** (`LANG`) o **minúsculas** (`lang`).
 - Se recomienda: **minúsculas** (`lang`)
 - **Valor** del atributos: sólo necesitan comillas si contienen espacios en blanco; las comillas pueden ser simples o dobles.
 - Se recomienda: siempre **comillas** (simples o dobles).

Atributos booleanos

- Sólo pueden tomar el valor verdadero o falso.
- Toma el **valor falso**: cuando no aparece en la etiqueta.
- Toma el **valor verdadero**: cuando aparece en la etiqueta:
 - Sin que se le asigne valor
 - Asignándoles un valor que puede ser el nombre del atributo o la cadena vacía

Ej. Los siguientes ejemplos son equivalentes:

<h1 hidden>

<h1 hidden="hidden">

<h1 hidden="">

Atributos globales

- Son atributos que se pueden especificar para todos los elementos HTML.

- ☐ id
- ☐ title
- ☐ lang
- ☐ translate
- ☐ dir
- ☐ class
- ☐ style
- ☐ accesskey
- ☐ contenteditable
- ☐ draggable
- ☐ dropzone
- ☐ hidden
- ☐ spellcheck
- ☐ tabindex

Espacios en blanco

- Al igual que los saltos de línea, en HTML los espacios en blanco no son significativos.
- Uno, dos o diez espacios en blanco se visualizan como un sólo espacio en blanco.

Referencias de caracteres

■ Necesarios para:

- Caracteres que **no existen en el código de caracteres** utilizado.

Ej. en ASCII no se pueden representar: @, á,

- Caracteres que tienen un **significado especial** en HTML.

Ej. el carácter < se interpreta como el inicio de una etiqueta.

- Caracteres que **no están disponibles en el teclado**.

Ej ® >? &>X£=?

■ Referencias de caracteres

- **Empiezan con & y terminan con ;**

- Se pueden escribir de tres formas:

- **Nombre:** &nombre;

- **Número decimal:** &#nnn;

- **Número hexadecimal:** &#Xnnn;

Referencias de caracteres

Carácter	Nombre	Descripción	Número
	<code>&nbspsp;</code>	Espacio en blanco que no se puede romper	<code>&#160;</code>
&	<code>&amp;</code>	Ampersand	<code>&#038;</code>
<	<code>&lt;</code>	Menor que	<code>&#060;</code>
>	<code>&gt;</code>	Mayor que	<code>&#062;</code>
©	<code>&copy;</code>	Copyrigh	<code>&#169;</code>
€	<code>&euro;</code>	Euro	<code>&#8364;</code>
£	<code>&pound;</code>	Libra	<code>&#163;</code>
"	<code>&quot;</code>	Dobles comillas	
'	<code>&apos;</code>	Comillas simples	
á, é, í, ó, ú	<code>&aacute;</code> <code>&eacute;</code> <code>&iacute;</code> <code>&oacute;</code> <code>&uacute;</code>	a, e, i, o, u con tilde aguda.	

Referencias de caracteres:

- *No-break space* (espacio en blanco que no se debe romper).
- Ejemplo: **Carlos V**
- No se debe utilizar para introducir espacios en blanco por una cuestión de estética: la presentación se debe controlar con CSS.

Validar

- Herramienta: <http://validator.w3.org/>
- **Empezar a corregir los primeros errores encontrados**, no por los últimos, que pueden ser consecuencia de los primeros.
- **Buena práctica** porque los documentos HTML válidos, es tienen más posibilidades de **visualizarse correctamente en diferentes navegadores**.

2. !DOCTYPE y metaelementos en HTML5

<!DOCTYPE html>

- Es el DOCTYPE que indica que es un documento de HTML5.

Necesario para asegurar que el navegador interprete el documento como HTML5 (por defecto puede interpretarlo de otro modo).

- Debe ser la primera línea del archivo: no debe haber espacios ni líneas antes de ella.

<html> ... </html>

- Elemento raíz del árbol de elementos.

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

```
</html>
```

- El único atributo que se necesita es **lang**.

<link rel="..." href="...">

- Permite incorporar, desde archivos externos: estilos, scripts, imágenes, iconos.
- Atributos
 - `<link rel="..." href="...">`
 - **rel**: relación entre el documento HTML y el archivo
 - **href**: ruta para cargar el archivo
 - **type**: opcional en HTML5
- **Ejemplo**: Incorporar una hoja de estilos:
 - `<link rel="stylesheet" href="css/estilo.css">`

<meta charset="UTF-8">

- **Valor de este atributo:**
 - Juego de caracteres del documento
 - Se puede escribir en mayúsculas o minúsculas.
- **Importante:** Se debe utilizar el mismo juego de caracteres en todos los ficheros de un sitio Web: HTML, PHP, CSS, JS, etc.
- **Cuál elegir:**
 - ☐ **UTF-8 sin BOM.** Permite utilizar texto en diferentes idiomas dentro del documento.
 - ☐ Si no se puede el anterior: **ISO-8859-1** (también llamado **Latin1**). Buen juego de caracteres para los idiomas del oeste de Europa.
 - ☐ Si se necesita el símbolo del euro: **ISO-8859-15** (**Latin9**)
 - ☐ Latin1 y Latin9 se diferencian en muy pocos caracteres.

3. Secciones

- En HTML5 se definen nuevas etiquetas para definir secciones.
- ¿Por qué se definen nuevas etiquetas?
 - Se utilizaban muchos div del tipo: `<div id="header">`
`<div id="nav">....`
 - Se han creado etiquetas específicas para esos usos.
 - De este modo, los navegadores y otros programas que rastrean la web pueden entender mejor el contenido del documento HTML.

- Etiquetas que definen “secciones”:
 - **body**
 - **article**, **section**
 - **nav**, **aside**
- Dentro de todas las “secciones” *puede* haber:
 - Títulos y subtítulos: **h1**, **h2**, **h3**, **h4**, **h5**, **h6**
 - Cabecera: **header**
 - Pie: **footer**
- Dentro de las “secciones” **article** y **body** *puede* haber:
 - Información de contacto: **address**

Esta etiqueta puede estar anidada dentro de otra etiqueta (div, header, footer), pero se refiere a su antecesor article o body.

<body> ... </body>

- Elemento de tipo «sección».
- Sólo puede haber un elemento <body> en la página.
- Contiene la parte visible de la página.

<article>

- Elemento de tipo «sección».
- Agrupa **contenido independiente**, que se puede extraer individualmente del documento y syndicar (ej. RSS) sin perder su significado.
- El ejemplo típico es una entrada de un blog.

Ejemplo

```
<article>
  <header>
    <h1>La primera clase</h1>
    <p>02/05/2005</p>
  </header>
  <p>....</p>
  <p>...</p>
  <footer>
    <a href="?comments=1">Mostrar
      comentarios...</a>
  </footer>
</article>
```

<section>

- Elemento de tipo «sección».
- Representa una **sección genérica de un documento o aplicación**.
- Una sección es un **grupo de contenido del mismo tema**. El tema de cada sección debería identificarse mediante un elemento de cabecera (h1-h6) que esté dentro de la sección.
- Ejemplos de secciones serían las zonas en las que se puede dividir una página web: introducción, news.
- Se aconseja utilizar article en lugar de sección cuando tendría sentido syndicar el contenido del elemento.

Ejemplo

```
<body >
  <h1 >Graduación</h1>
  <section>
    <h2>Programa de actos</h2> >
    <ul>
      <li>Palabras de..</li>
      <li>...</li>
    </ul>
  </section>
  <section>
    <h2>Graduados</h2> >
    <ul>
      <li>Susana R.</li>
      <li>...</li>
    </ul>
  </section>
</body>
```


<nav>

- Elemento de tipo «sección».
- Contiene los **enlaces de la navegación principal** (dentro del documento o a otras páginas).
- También se puede utilizar para **navegación secundaria**.
- Puede haber **varios** elementos <nav> dentro de una página.
- No todos los grupos de enlaces tienen que estar en un elemento nav, sólo los que forman bloques de navegación principal. Por ejemplo, en el footer suele haber una lista corta de enlaces que no es necesario que formen parte de un <nav>.
- El elemento <nav> puede formar parte del <header> o no.
- Si contiene una lista de enlaces, en general se recomienda utilizar el elemento ul.

Ejemplo

```
<nav>
  <ul>
    <li><a href="/">Home</a></li>
    <li><a href="/events">
      Events</a></li>
    ...
  </ul>
</nav>
```

<aside>

- Elemento de tipo «sección».
- Contenido: relacionado tangencialmente con el contenido que está a su alrededor, y que podría tener sentido separado de ese contexto.
- Se puede utilizar para efectos tipográficos, anuncios, agrupar elementos nav, etc.

Ejemplo

Utilizar un elemento <aside> para una cita, dentro de un artículo más largo.

```
<p>Mi trabajo...</p>
```

```
<aside>
```

```
    <q>La gente me suele preguntar cómo me  
    divierto cuando no estoy trabajando... </q>
```

```
</aside>
```

```
<p>....</p>
```

<h1> ... <h6>

- Título de un elemento de tipo “sección” (**article**, **section**, **nav**, **aside** o **body**).
- Se recomienda seguir el orden lógico: h1, h2, h3, ...
- En cada elemento de tipo “sección”, se puede empezar desde h1.

Sin embargo se recomienda que en cada página HTML haya sólo un h1, y que los títulos definan la estructura de contenidos de la página.

<header>

- Cabecera de un elemento de tipo sección (**section**, **article**, **aside**, **nav** o **body**).
- Contenido introductorio. Normalmente contiene un grupo de ayudas introductorias o de navegación.
- No siempre contiene elementos de cabecera como h1-h6. Puede contener, por ejemplo, un formulario de búsqueda, o la tabla de contenidos de la sección.

Ejemplo

```
<article>
  <header>
    <h1>La primera clase</h1>
    <p>02/05/2005</p>
  </header>
  <p>....</p>
  <p>...</p>
  <footer>
    <a href="?comments=1">Mostrar
      comentarios...</a>
  </footer>
</article>
```

<footer>

- Pie de un elemento de tipo sección (**section**, **article**, **aside**, **nav** o **body**).
- Normalmente contiene información sobre la sección, como quién lo escribió, enlaces a otros documentos relacionados, datos de copyright.
- Puede contener otras secciones.

Ejemplo

```
<footer>
<!-- footer del sitio -->
  <nav>
    <p> <a href="..">credits</a> -
      <a href="...l">terms</a> -
      <a href="..">index</a>
    </p>
  </nav>
  <p>copyright © 2009 ...</p>
</footer>
```

<address>

- Información de contacto de un elemento **article** o de un elemento **body**.
- Se puede incluir en un footer.
- No debe utilizarse para direcciones si no son direcciones de contacto.

Ejemplo

```
<footer>  
  <address>  
    Para más detalles, contactar con <a  
      href="mailto:js@example.com">John  
      Smith</a>.  
  </address>  
  <p><small>© copyright 2038  
    Example Corp.</small></p>  
</footer>
```

Compatibilidad con navegadores no HTML5

- Las nuevas etiquetas HTML5 no son reconocidas por los navegadores que no soportan HTML5.
- Soluciones:
 - Con estilos:

```
section, article, .. {  
    display: block;  
}
```
 - Con JavaScript

https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Using_HTML_sections_and_outlines#Using_HTML5_elements_in_non-HTML5_browsers

4. Agrupar contenido

- p
- hr
- pre
- blockquote
- ol
- ul
- li
- dl
- dt
- dd
- figure
- figcaption
- div
- main

<https://www.w3.org/TR/html52/grouping-content.html#grouping-content>



ol: listas ordenadas

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

1. Elemento 1
2. Elemento 2
3. Elemento 3
4. Elemento 4
5. Elemento 5

Las listas se pueden **anidar**.



ul: listas desordenadas

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

- Elemento 1
- Elemento 2
- Elemento 3
- Elemento 4
- Elemento 5

Las listas se pueden **anidar**.

<dl>, <dt>, <dd>

dl: listas de definiciones

```

<dl>
  <dt>Director</dt>
  <dd>Es el responsable de...</dd>
  <dt>Jefe de estudios</dt>
  <dd>Es el responsable de...</dd>
  <dd>Además...</dd>
  <dt>Secretario</dt>
  <dd>Es el responsable de...</dd>
  <dt>Conserjes</dt>
  <dt>Personal administrativo</dt>
  <dd>Son los responsables de...</dd>
</dl>

```

Director
Es el responsable de...

Jefe de estudios
Es el responsable de...
Además...

Secretario
Es el responsable de...

Conserjes

Personal administrativo
Son los responsables de...

Las listas <dl> pueden anidarse dentro de otras, pero no pueden contener otras dentro de ellas.

<figure> y <figcaption>

Se utilizan para encapsular una figura como un sólo item.

Puede contener un título para la figura.

<figure>

<figcaption>

</figcaption>

</figure>

5. Semántica a nivel de texto

Etiquetas *in-line* que permiten definir semántica:

- | | | |
|----------|-------------|--------|
| ■ a | ■ code | ■ rb |
| ■ em | ■ var | ■ rt |
| ■ strong | ■ samp | ■ rtc |
| ■ small | ■ kbd | ■ rp |
| ■ s | ■ sub y sup | ■ bdi |
| ■ cite | ■ i | ■ bdo |
| ■ q | ■ b | ■ span |
| ■ dfn | ■ u | ■ br |
| ■ abr | ■ mark | ■ wbr |
| ■ data | ■ ruby | |
| ■ time | | |

6. Editar contenido

- ins
- del

7. Contenido embebido

- Permiten introducir en el documento
 - Otro recurso o archivo: ej., imagen, vídeo, página web
 - Otro contenido: ej., svg.
- **Todas estas etiquetas son elementos in-line** (*phrasing content*).
- Etiquetas que permiten introducir “contenido embebido” : **img**, **iframe**, **embed**, **object**, **audio**, **video**, **canvas**, **math** y **svg**.
- En esta presentación, veremos **img** e **iframe**.

Atributos width y height

- Ancho de la imagen en píxeles: **width="número"**
- Alto de la imagen en píxeles: **height="número"**
- Utilizar estos atributos hace que la página se cargue más deprisa.
- Utilizar el tamaño real de la imagen, para que el navegador no tenga que escalarla.

■ Texto alternativo:

- ☐ Texto que se muestra si no se puede cargar una imagen.
- ☐ Texto utilizado por los lectores de pantalla.

■ Valor del texto alternativo:

- ☐ Una breve descripción de lo que hay en la imagen.
- ☐ Si una imagen no añade ningún significado al texto se recomienda alt=""

Imágenes decorativas

- Se recomienda mover las imágenes puramente decorativas al CSS (*background-image*).
- Ventajas:
 - ☐ Documento más limpio y accesible.
 - ☐ Más fácil hacer cambios en el aspecto.

<iframe src="" width="" height="" >
contenido alternativo
</iframe>

- Permite embeber en un documento HTML otro documento HTML separado u otro recurso.
- A veces se utiliza para contenido de terceros, como anuncios interactivos u otros widgets, garantizando así que no interfieren con el contenido y scripting del resto de la página.

```
<iframe src="lista.html" width="400" height="200">
```

Tu navegador no soporta inline frames.

Lee la [Lista](lista.html)

```
</iframe>
```

8. Enlaces

■ Sintaxis

`texto o elemento`

■ ¿Qué puede ir entre las etiquetas <a> y

- HTML 4.01: Elementos inline.
- HTML5: Cualquier elemento, incluso elementos de bloque.

■ Tipos de referencias "URL":

- Cómo indicar el documento.
 - **Referencias absolutas:** en cualquier sitio Web.
 - **Referencias relativas:** en el propio sitio Web.
- Cómo indicar un lugar concreto de un documento.
- Otras referencias (correo electrónico, ficheros para descarga).

- Para referenciar un documento de otro sitio Web
- Empiezan con el protocolo (http://), nombre de dominio y ruta si es necesaria.

```
<a href="http://www.w3c.org">w3c</a>
```


- Para referenciar un documento de tu propio sitio.
- Se escribe la página web, y delante la ruta relativa.

`foto`

- ☐ En la carpeta padre

`foto`

- ☐ En una subcarpeta

`foto`

- ☐ Relativo a la raíz del sitio

`foto`

- **PÁGINA DESTINO:** `id="nombre"`

```
<h1 id="inicio">TITULO</h1>
```

- **PÁGINA DONDE SE REFERENCIA:** `#nombre`

- ☐ Enlace en la misma página Web

```
<a href="#inicio">Ir a inicio</a>
```

- ☐ Enlace a otra página del sitio

```
<a href="glosario.html#inicio">Ir al glosario (inicio)</a>
```

- ☐ Enlace a otra página de otro sitio

```
<a href="http://w3c.org/glosario.html#inicio">Ir al  
glosario (de w3c, inicio)</a>
```


- **Atributo target: definir en qué ventana se abre la página enlazada**

target="_blank" En una pestaña nueva del navegador o en una nueva ventana.

target="_self" En la misma ventana.

target="_parent" En la ventana padre (marcos).

target="_top" En la ventana superior del conjunto de marcos.

target="nombre" En una pestaña nueva del navegador o en una nueva ventana.

- Nombre es cualquier nombre que no empiece por subrayado (_)
- Todos los documentos que se abran indicando ese target, se abren en la misma ventana.

- **Consejo:** Puede ser mejor no utilizar este atributo.

- ☐ El usuario se pierde, es más difícil que vuelva.
- ☐ Si accede desde dispositivos móviles, es más incómodo.
- ☐ Si accede con dispositivos lectores, puede que no lo sepan interpretar.

``

- Enlace a un correo

- Utilizar el protocolo mailto:

`...`

- Cuidado:

- Proporcionar un correo electrónico en un documento HTML es arriesgarse a recibir spam.
 - Algunas soluciones: escribirlo de modo que las personas puedan leerlo, pero los robots no:
 - pepe at perez.com
 - imagen

■ Enlaces a teléfonos

- Aprovechar que los smartphones que se utilizan para acceder a Internet, permiten hacer llamadas telefónicas!

+18050505

- Sólo funciona desde dispositivos móviles (excepto iPad y iPod Touch). No funciona desde equipos de escritorio.
- Consejos
 - Escribir el número internacional
 - Repetir el número entre las etiquetas <a>, por si no funciona el enlace

9. Tablas

- Se utilizan para presentar contenidos que se organizan en forma de tabla, como calendarios, estadísticas, etc.

ALUMNO	NOTA
Pepe	7,65
María	8,54
Jordi	5,55

```
<table>
  <tr>
    <th>Producto</th>
    <th>Precio</th>
  </tr>
  <tr>
    <td>Aceituna</td>
    <td>2,60</td>
  </tr>
  <tr>
    <td>Almendra</td>
    <td>1,20</td>
  </tr>

  <tr>
    <td>Cacahuete</td>
    <td>1,60</td>
  </tr>
  <tr>
    <td>Total</td>
    <td>5,40</td>
  </tr>
</table>
```

Producto	Precio
Aceituna	2,60
Almendra	1,20
Cacahuete	1,60
Total	5,40

Etiquetas td y th

- Son las etiquetas en las que va el contenido.
- Contenido de una celda td o th: puede ser cualquiera: texto, gráfico, otra tabla, etc.

Etiquetas th

- Sustituye a la celda <td> para celdas de cabecera.
- Se pueden utilizar para cualquier fila, aunque se recomienda para las primeras y últimas.
- También puede ser una columna.
- Los navegadores suelen presentar su contenido en **negrita**.

- Combinar celdas: atributos **colspan** y **rowspan** de `<td>` y `<th>`
- A estos atributos se les asigna un valor entero que especifica cuántas columnas o filas ocupa la celda.

```

<table>
  <tr>
    <th colspan="2">Precios de venta</th>
  </tr>
  <tr>
    <th>Producto</th>
    <th>Precio</th>
  </tr>
  <tr>
    <td rowspan="2">Aceituna</td>
    <td>2,60</td>
  </tr>
  <tr>
    <td>1,20</td>
  </tr>
  <tr>
    <td>Cacahuete</td>
    <td>1,60</td>
  </tr>
  <tr>
    <td>Total</td>
    <td>5,40</td>
  </tr>
</table>

```

Precios de venta	
Producto	Precio
Aceituna	2,60
	1,20
Cacahuete	1,60
Total	5,40

Accesibilidad: atributo scope de la etiqueta <th>

Syntax

```
<th scope="col|row|colgroup|rowgroup">
```

Attribute Values

Value	Description
col	Specifies that the cell is a header for a column
row	Specifies that the cell is a header for a row
colgroup	Specifies that the cell is a header for a group of columns
rowgroup	Specifies that the cell is a header for a group of rows

Etiquetas **thead**, **tbody**, **tfoot**

- Se utilizan para agrupar filas.
- ¿Por qué agrupar filas?
 - Algunos **navegadores especializados** pueden repetir las cabeceras o los pies cuando una tabla ocupa varias páginas.
 - Para **dar estilo por grupos de filas**.

```
<table>
  <thead>
    <tr>
      <th>Producto</th>
      <th>Precio</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Aceituna</td>
      <td>2,60</td>
    </tr>
    <tr>
      <td>Almendra</td>
      <td>1,20</td>
    </tr>

    <tr>
      <td>Cacahuete</td>
      <td>1,60</td>
    </tr>

  </tbody>
  <tfoot>
    <tr>
      <td>Total</td>
      <td>5,40</td>
    </tr>
  </tfoot>
</table>
```

Título de las tablas: <caption>

- ☐ Utilizar este elemento para dar un título o breve descripción que se muestre junto a la tabla.
- ☐ Si se utiliza, debe ser el primer elemento dentro del elemento <table>.

Agrupar columnas con `<colgroup>` y `<col>`

- Para **dar estilo** a columnas o grupos de columnas.
- Etiquetas:
 - Grupo de columnas: `<colgroup>` `</colgroup>`
 - Columnas dentro del grupo de columnas: `<col>`
- El atributo **span** (de `<colgroup>` y `<col>`) indica el número de columnas.
- Si se utiliza esta etiqueta, tiene que aparecer al principio, después de `<table>` y `<caption>` (si existe).

```
<table>
  <colgroup>
    <col span="2" style="background-color:red">
    <col style="background-color:yellow">
  </colgroup>
  <tr>
    <th>ISBN</th>
    <th>Title</th>
    <th>Price</th>
  </tr>
  <tr>
    <td>3476896</td>
    <td>My first HTML</td>
    <td>$53</td>
  </tr>
  <tr>
    <td>5869207</td>
    <td>My first CSS</td>
    <td>$49</td>
  </tr>
</table>
```

ISBN	Title	Price
3476896	My first HTML	\$53
5869207	My first CSS	\$49

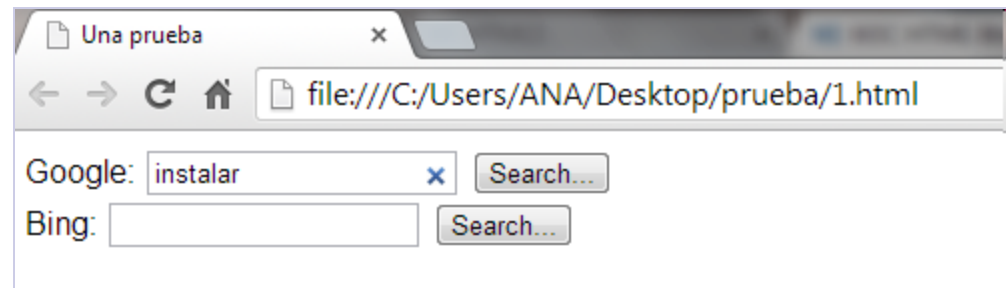
10. Formularios

- `<form action="..." method="..." > ... </form>`
- Contiene los elementos de control de un formulario.
- ¡No anidar!
- Atributos
 - `action="URL"`
 - URL de la aplicación o script.
 - `method="post|get"`
 - Método utilizado para enviar los datos del formulario.
 - **post**: el navegador envía una petición separada al servidor con cabeceras especiales seguidas de los datos. Más seguro.
 - **get**: el navegador envía los datos al servidor en la URL.
 - `enctype="..."`
 - Método de codificación. Normalmente `application/x-www-form-urlencoded` (default) o `multipart/form-data`

Este ejemplo muestra dos formularios de búsqueda:

```
<form action="http://www.google.com/search" method="get">  
<label>Google: <input type="search" name="q"></label>  
<input type="submit" value="Search...">  
</form>
```

```
<form action="http://www.bing.com/search" method="get">  
<label>Bing: <input type="search" name="q"></label>  
<input type="submit" value="Search...">  
</form>
```



Accesibilidad

Diseñar los formularios para los usuarios que no pueden utilizar navegadores visuales.

Tres etiquetas mejoran la accesibilidad de los formularios, haciendo conexiones semánticas entre los componentes de un formulario.

Son: **label**; **fieldset** y **legend**

<label for="...">
</label>

- Permite asociar un texto (etiqueta) con un control de formulario.
- Dos formas de hacerlo:
 - Utilizando el atributo **for** en el elemento **label**, y el atributo **id** en el elemento de control

```
<label for="form-nombre">Nombre:</label>
<input type="text" name=nombre id="form-nombre">
```
 - Colocando el control dentro del elemento **label**.

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

<fieldset>
<legend>
</legend>

...

</fieldset>

- Con **<fieldset>** se agrupan elementos de control agrupados bajo un nombre común. Y con **<legend>** se les da un título.
- Atributos
 - ☐ **name="..."**
Nombre del elemento.
 - ☐ **form="nombre_formulario"**
Para indicar explícitamente el nombre de su formulario.
 - ☐ **disabled**
Hace que todos los elementos hijos de fieldset, excluidos los hijos del primer elemento legend, estén desactivados
- Los elementos **fieldset** se pueden anidar.

```

<fieldset name="clubfields" disabled>
<legend> <label>
    <input type=checkbox name=club
        onchange="form.clubfields.disabled = !checked">
    Use Club Card </label>
</legend>
<div><label>Name on card: <input name=clubname required>
    </label></div>
<div><label>Card number: <input name=clubnum required
    pattern="[-0-9]+">
    </label></div>
<div><label>Expiry date: <input name=clubexp type=month>
    </label></div>
</fieldset>

```

Una prueba

file:///C:/Users/ANA/Desktop/prueba/1.html

☒ Use Club Card

Name on card:

Card number:

Expiry date: de

Atributos comunes a la mayoría de los controles

- name
- value
- size
- maxlength
- disabled
- readonly

Atributo name

- Es el **nombre de la variable** donde se almacenará el valor recogido en ese elemento de control.
- Debe ser **única en el formulario**.
- Atributo **obligatorio** en **casi todos** los controles de un formulario, **excepto en submit y reset** (que lo pueden llevar, pero no es obligatorio porque tienen una función dentro del formulario diferente de recoger datos).
- **Ejemplo** (de uno de los dos formularios del ejemplo anterior):

```
<label>Google: <input type="search" name="q"></label>
```

Atributo value

- ☐ Valor de la variable.
- ☐ En los campos de tipo texto, este valor aparecerá en el campo cuando se carga o cuando se reinicializa el formulario.

`value="valor"`

Atributo size

- Es el número de caracteres que el navegador permite ver cuando se edita el contenido de un elemento de control.

Atributo maxlength

- Es el número máximo de caracteres que puede tener la variable.

Atributo disabled

- Es un atributo booleano.
- Se utiliza para deshabilitar el contenido (la posibilidad de modificarlo).
- Si se utiliza:
 - ☐ En un control: deshabilita el control.
 - ☐ En un elemento fieldset: deshabilita todos los elementos del grupo (excepto los descendientes del elemento legend, si hay alguno).

Atributo readonly

- Es un atributo booleano.
- Se utiliza para evitar que un usuario pueda editar el control.

<input type="..." name="..." >

Algunos usos de la etiqueta <input>

type="..."	descripción
text	Permite introducir un texto, es el tipo por defecto.
password	Funciona como un control de entrada de texto, sólo que no se muestra el texto que vamos escribiendo. En su lugar veremos unos asteriscos u otro símbolo. No ofrece seguridad (no cifra)
file	Muestra un botón Seleccionar o Examinar para poder elegir el archivo que se quiere enviar. Requiere que el formulario envíe los datos mediante el método "post" y utilizando el atributo enctype="multipart/form-data". Permite enviar múltiples archivos, utilizando el atributo multiple.
hidden	Mantiene un dato oculto a la vista pero se pasa al servidor junto con el resto de datos del formulario. Campo adecuado para pasar información de una página a otra. Además, se puede utilizar en los scripts como un dato más.

<input>: tipos HTML5

■ Entradas de texto especiales

- ☐ `type="search"`
- ☐ `type="email"`
- ☐ `type="tel"`
- ☐ `type="url"`

■ Entradas numéricas

- ☐ `type="number"`
- ☐ `type="range"`
- ☐ atributos `min`, `max` y `step`

■ Entradas de fechas y hora

- ☐ `type="date"`
- ☐ `type="time"`
- ☐ `type="datetime"`
- ☐ `type="datetime-local"`
- ☐ `type="month"`
- ☐ `type="week"`

■ Entradas para seleccionar un color

- ☐ `type="color"`

<input>: tipos HTML5 y compatibilidad

- ¿Qué ocurre con los navegadores que no reconocen los tipos de HTML5?
 - ☐ Los ignora
 - ☐ Asigna el tipo por defecto: **type="text"**
- Mejora progresiva: diseñar para que las páginas se puedan ver con navegadores muy avanzados y menos avanzados.

<input type="radio">

- Un grupo controles de tipo "radio" tienen **asociada la misma variable** (atributo name).

Sólo puede haber uno seleccionado: a la variable se le asignará el valor (value) del elemento seleccionado

- Atributos

- ☐ type="radio"
- ☐ name="variable"
- ☐ value="valor"
- ☐ checked

```
<p>Selecciona la edad</p>
```

```
<ul>
```

```
  <li><input type="radio" name="edad" value=15 checked>15</li>
```

```
  <li><input type="radio" name="edad" value=20>20</li>
```

```
  <li><input type="radio" name="edad" value=30>30</li>
```

```
</ul>
```

<input type="checkbox">

- Control que puede tener dos estados: activado o no activado.
- Atributos
 - **type="checkbox"**
 - **checked**

<input type="submit", "reset" y "button">

- `<input type="submit" value="...">`
 - Envía los datos de un formulario al servidor.
- `<input type="reset" value="...">`
 - Reinicia los controles de un formulario a sus valores por defecto.
- `<input type="button" value="...">`
 - Crea un botón genérico sin una función concreta. Se le suele dar funcionalidad con JavaScript.
- Atributos
 - `value="..."` Texto del botón

<textarea> ... </textarea>

- Para introducir varias líneas de texto.
- El texto que se escriba entre las dos etiquetas será el valor por defecto.
- Atributos
 - **rows="n"** número de líneas de texto
 - **cols="n"** número de columnas (caracteres) de texto

<button>...</button>

- `<button> ... </button>`
- Representa un **botón etiquetado con el texto** que hay entre las dos etiquetas.
- Atributos
 - `type="submit|reset|button|menu"`
 - **submit** Envía el formulario (default)
 - **reset** Reinicia el formulario
 - **button** No hace nada
 - **menu** Muestra un menú
 - Si el tipo es "submit", el navegador realizará la **validación de restricciones**.

<select>... </select>

<select>...</select>

<option>...</option>

<optgroup>...</optgroup>

- Menús desplegables
- Menús con scroll
- Agrupar: <optgroup label="...">
 - Opcional; aparece al principio del grupo.

<datalist>...</datalist>

- Muestra una lista de valores posibles para un elemento <input>.
- A diferencia de <select>, el usuario puede introducir un valor diferente a los proporcionados en la lista.

```
<input name="estudios" type="text" list="tipos_est">
<datalist id="tipos_est">...
    <option value="primaria">
    <option value="secundaria">
    <option value="universidad">
</datalist>
```

Nuevos atributos

- **autocomplete="on" o "off"**
 - ☐ mostrar los textos de entradas anteriores como posibles valores
- **novalidate**
 - ☐ atributo para la etiqueta form, permite enviar un formulario sin validar
- **formnovalidate**
 - ☐ atributo para la etiqueta submit, permite enviar un formulario sin validar
- **placeholder**
 - ☐ Muestra un texto para ayudar al usuario a rellenar el formulario
- **required**
 - ☐ El formulario no se debe enviar si el campo está vacío

Nuevos atributos (2)

■ **multiple**

- ☐ Permitir múltiples entradas en un mismo campo. Los valores se escriben separados por comas. Sólo lo permiten algunos controles como **email** y **file**.

■ **autofocus**

- ☐ coloca el foco

■ **pattern**

- ☐ Define un patrón para validar una entrada
- ☐ Ej. `<input pattern="[0-9]{5}" name="cpostal" title="Inserte código postal">`

■ **form**

- ☐ Permite declarar elementos de un formulario fuera de la etiqueta form

Otras etiquetas HTML5

progress

meter

output

keygen

Referencias

- Especificaciones oficiales de HTML:
 - HTML 4.01: <http://www.w3.org/TR/html4/>
 - XHTML 1.0: <http://www.w3.org/TR/xhtml1/>
 - HTML5: <https://www.w3.org/TR/html52/> <http://www.w3.org/TR/html5/>
- Libros Web: <http://librosweb.es/>
- Especificaciones de CSS: <https://www.w3.org/Style/CSS/specs.en.html>
 - Especificaciones de CSS2 (en castellano):
<http://www.sidar.org/recur/desdi/traduc/es/css/cover.html#minitoc>
- Compatibilidad de los navegadores: <http://caniuse.com/>
- Web technology for developers
<https://developer.mozilla.org/en-US/docs/Web>
- HTML Training de W3C (inglés, algunos materiales en castellano):
<http://www.w3.org/community/webed/wiki/HTML/Training>.
- Tutoriales de HTML: www.w3schools.com
- WHATWG: <http://www.whatwg.org/>