UD1_ Aplicacións web e linguaxes de marcas: HTML e CSS

ACTIVIDADE 4: Más HTML5

Índice

1. Organizar el contenido con etiquetas de bloques de contenido

div

El elemento `div` es un elemento utilizado desde las primeras versiones de HTML que permite agrupar contenido y, combinado con CSS permite crear "capas", entendiendo por capas elementos contenidos en otro elemento que después podremos colocar en cualquier parte de la página.

Por defecto, los navegadores no dan formato al contenido de un elemento `div`.

1.1 Etiquetas estructurales de HTML5

La idea en los nuevos estándares de HTML (especialmente en HTML5) es que los elementos HTML más que para formato, sirvan para dar valor semántico al contenido. Es decir, para indicar qué tipo de contenido es. Por ejemplo la etiqueta `<p>` sirve para crear un párrafo, con `<abbr>` definimos una abreviatura, etc.

La idea es marcar la semántica con HTML y el formato con las hojas de estilo CSS.

La parte visual no lo es todo. Utilizamos diferentes fuentes y colores para llamar la atención de los usuarios de las partes más importantes del contenido, como el menú de navegación y sus correspondientes enlaces, pero ¿qué ocurre con los usuarios discapacitados visuales[1] que no pueden distinguir los colores llamativos o las fuentes de mayor tamaño?

En nuestro código HTML, podemos crear secciones de contenido basadas en su funcionalidad — usando elementos que representen las diferentes secciones del contenido — , de forma que las tecnologías de ayuda y los lectores de pantalla puedan reconocer esos elementos y asistir en tareas como "*encontrar el menú de navegación*", o "*encontrar el contenido principal*".

A este respecto hay una serie de elementos aparecidos con HTML5 que no dan ningún formato al texto pero permiten remarcarlo dándole un significado y además permitiendo posteriormente que ese contenido tenga un formato especial mediante CSS.

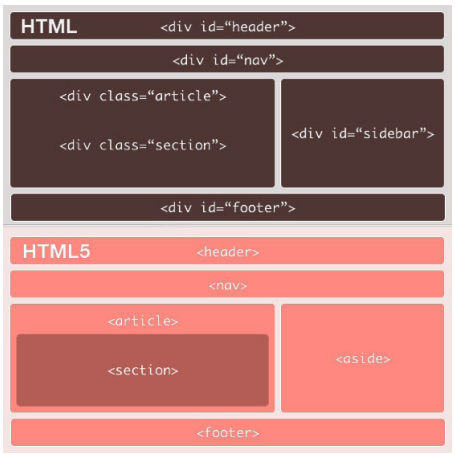
En HTML4 no existen muchas herramientas para definir el contenido de una página Web. Las más comunes son `p` para identificar un párrafo y `div` para una sección de contenido. No hay ninguna etiqueta que describa su contenido.

La mayoría de las webs tienen un formato común, formado por elementos como cabecera, pie, navegadores, etc. Tomando en cuenta la utilización que damos a algunos elementos `div` con identificadores muy comunes, los desarrolladores crearon *etiquetas estructurales que permiten definir las partes básicas de una página y estructurar su contenido*.

En la imagen siguiente podemos observar un layout típico que utiliza *`divs`* con atributos *`id`* y/o `class` para estructurarlos. Estos contienen un *header*, *footer* y una barra

[1] *Se estima que en todo el mundo hay cerca de **300 millones de personas** que son **daltónicas** y **253 millones con ceguera o discapacidad visual grave.***

de navegación horizontal debajo del header. El contenido principal contiene un *article* y el *sidebar* se encuentra a la derecha.



HTML5 introduce nuevos elementos para representar cada una de estas secciones, con lo cual en el ejemplo de la imagen, los elementos `div` pueden ser reemplazados con los nuevos elementos: `header`, `nav`, `section`, `article`, `aside`, y `footer`. (también tenemos la etiqueta `main`, aunque esta no se considere una etiqueta de sección sino de bloque)

Al utilizar etiquetas específicas para tipos de contenido específico, se mejora la legibilidad y facilidad de reutilización tanto del código HTML como de las hojas de estilo.

La etiqueta `div` se mantiene en la recomendación HTML 5, pero reservada ya al resto de agrupaciones no consideradas por las etiquetas anteriores, es decir, con fines principalmente decorativos.

header

Representa la cabecera de la página o de una sección. Las cabeceras, además del título, pueden incluir un logo, subtítulos o información adicional.

Esto no tiene que ver con el elemento `head`, se trata de una serie de párrafos que se marcan de esta forma para indicar que pertenecen a la cabecera de la página. Realmente una página puede tener varios elementos header, al nivel de la etiqueta `body` indica que su contenido es la cabecera de la página completa (por ejemplo donde se muestra el título general, logotipo, nombre de la empresa). Pero dentro de una etiqueta, por ejemplo, de la etiqueta `article` indicaría que su contenido es la cabecera del artículo.

footer

Contiene la información que suele estar en el pié de un documento, autor, copyright... independientemente si está situada en el pié del documento, a un lado o donde sea. Al igual que `header` también puede usarse con las etiquetas `article` o `section` si estas lo requieren.

section

Es un elemento que permite dividir en diferentes partes o secciones un documento. Identifica claramente un bloque de contenidos. Agrupa una parte de documento con contenido de temática similar, por ejemplo una introducción o un subapartado. En un blog, sería la zona donde están todos los posts. En un video de youtube, habría un section para el video, uno para los datos del video, otro para la zona de comentarios.

Una sección puede contener otras secciones

article

Define un bloque de contenido que tiene sentido por sí mismo en la página. En el home de un blog, cada post sería un `article`. En un post del blog, el post y cada uno de sus comentarios sería un `article`.

Cuándo utilizar div, section y article

A veces nos surgen dudas a la hora de diferenciar `div`, `section` y `article`. Básicamente, la diferencia está en la carga semántica que le queramos dar al contenido. ¿Cuándo usamos un div, un section o un article?

Usaremos `article` para representar cualquier información que tenga sentido propio por si misma y sea propensa a la reutilización, por ejemplo, mediante sindicación. Un ejemplo: un post de un blog, el resumen de ese post en una portada, un hilo en un foro, un comentario…

Usaremos `section` para agrupar contenido relacionado. Puede servirnos para enmarcar un conjunto de artículos, o incluso nuevas y diferentes secciones, o artículos que se dividan en secciones… la variedad es infinita.

`div` no tiene ninguna intención semántica, es un simple "agrupador" de contenido que nos ayudará a pintarlo con CSS.

aside

Con el elemento `aside` se representa una parte de la página que abarca un contenido tangencialmente relacionado con el contenido que lo rodea, pero que no pertenece a ella, por lo que se le puede considerar un contenido independiente. Tiene la misma función que las barras laterales de referencia o que las secciones laterales de los libros y artículos.

Este elemento puede utilizarse para efectos tipográficos, barras laterales, elementos publicitarios, para grupos de elementos de la navegación, u otro contenido que se considere separado del contenido principal de la página.

nav

La etiqueta `nav` define una sección de vínculos de navegación. No todos los enlaces de un documento deben ser un elemento `nav`. La función de esta nueva etiqueta consiste en identificar los grupos de enlaces que, una vez agrupados, componen la navegación de la página. Algunos contenidos que se pueden agrupar gracias a esta etiqueta son: los enlace principales, los enlaces que remiten a datos como Anterior y Siguiente o las migas de pan.

Navegadores, tales como lectores de pantalla para usuarios con discapacidad, puede utilizar este elemento para determinar si se omite la representación inicial de este contenido.

1.2 Etiquetas que dotan de mayor semántica a la página

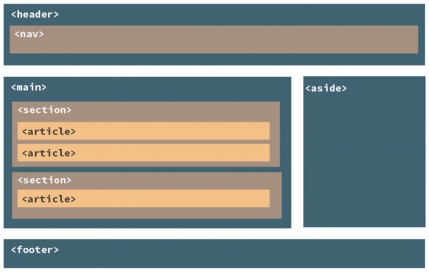
Existen otras etiquetas que dotan de mayor semántica a la página web. Estas son:

main

El propósito principal de esta etiqueta es desde un punto de vista de ***accesibilidad***, ya que ayuda a que los screen readers (lectores de pantalla) y otras tecnologías asistenciales puedan identificar donde comienza el contenido principal de la página y donde termina.

Según la especificación: el elemento `main` representa el contenido principal del cuerpo (`body`) de un documento.

Una característica importante a tener en cuenta sobre este elemento, es que puede ser usado *una sola vez por página*, algo que parece bastante obvio si pensamos su propósito. Usar más de un elemento `main`, hará que nuestro HTML sea inválido para la W3C.



Este elemento no puede ser hijo de ninguno de los siguientes: `header`, `nav`, `article`, `aside` y `footer`.

Así, introduciendo el elemento main, una estructura de nuestra página podría ser la que se ve en la imagen.

figure

La etiqueta `figure` identifica una imagen (o un objeto relacionado con el texto) y su descripción como parte de un conjunto. Este elemento puede ser usado para ilustraciones, diagramas, fotos, listas de código, etc, que están referidos desde el contenido principal del documento.

a etiqueta `figcaption` identifica el texto que acompaña al objeto. Por ejemplo:

```
<figure>
   <figcaption>Fig.10 Etiquetas obsoletas en HTML5</figcaption>
   <img alt=”Etiquetas obsoletas en HTML5” src=”etiquetas_obsoletas.png” />
</figure>
```

A continuación vemos otro ejemplo donde se asocia una misma leyenda a dos imágenes:

```
<figure>
   <figcaption>Fig.11 Lenguajes de marcas</figcaption>
   <img alt=”HTML5” src=”html5.png”/>
   <img alt=”XML” src=”xml.png”/>
</figure>
```

mark

Sirve para resaltar partes del texto.

```
<p>Podemos dar <mark>énfasis</mark> al texto con la etiqueta <mark>mark</mark></p>
```

time

Permite identificar un contenido como medida de tiempo. Este elemento puede ser utilizado como una forma de codificar las fechas y horas de una forma legible por máquina, de modo que, por ejemplo, los agentes de usuario pueden ofrecer para añadir recordatorios de cumpleaños o eventos programados en el calendario del usuario, y los motores de búsqueda puede producir resultados más inteligentes.

A continuación se muestran dos ejemplos:

```
<p>Las clases empiezan a las <time>8:20</time> de la mañana.</p>

<p>Mi cumpleaños es el <time datetime="2010-01-06">día de reyes.</time>.</p>
```

dfn

Se usa para identificar un término que va a ser definido a continuación.

```
<p>
     El <dfn>HTML5</dfn> es un lenguaje de marcas empleado para estructurar y
     presentar contenido en la WWW.
</p>
```

1.3 Crear elementos expandibles

Antes de HTML5, para crear un elemento expandible necesitábamos recurrir a código Javascript. Ahora es posible hacerlo de forma sencilla utilizando la etiqueta `details`. Utilizando esta etiqueta junto con `summary` podemos crear un elemento que al desplegarse revela determinada información oculta:

La etiqueta `summary` contiene el texto que será visible cuando el elemento esté contraído, y a continuación va el contenido que se mostrará al expandir el elemento (en nuestro ejemplo un párrafo), que puede ser casi cualquier cosa, incluyendo imágenes, tablas, vínculos, etc.

```
<details>
   <summary>Más información</summary>
   <p>Este texto se muestra o esconde al pinchar en "Más información" </p>
</details>
```

Que se visualizaría: ▶ Más información

Y la hacer clic sobre él muestra la información oculta:

▼ Más información

Este texto se muestra o esconde al pinchar en "Más información"

Si queremos que por defecto se muestre abierto añadiremos el atributo `open`.

```
<details open="open"> ...
```

Haz la tarea 1 donde practicarás la estructura de páginas usando las etiquetas explicadas anteriormente.

1.4 Tablas

1.4.1 Creación de tablas simples

Una tabla es un conjunto de datos presentados en forma de filas y columnas.

En HTML cada tabla está asociado a un elemento `<table>`, dentro de este elemento se indican las filas mediante el elemento `<tr>` y dentro de cada fila se indican las celdas mediante elementos `<td>` cuando contienen datos, y `<th>` (siglas de Table Headings) cuando estas celdas sirven de cabecera de fila o de columna (los navegadores las mostrarán resaltadas de algún modo.

A continuación vemos un ejemplo de una tabla:

```
<table>
    <tr>
        <th></th>
        <th>Lunes</th>
        <th>Martes</th>
        <th>Miércoles</th>
        <th>Jueves</th>
        <th>Viernes</th>
    </tr>
    <tr>
        <td>8:20</td>
        <td>FOL</td>
        <td>Bases de datos</td>
```

```
            <td>Sistemas operativos</td>
            <td>Programación</td>
            <td>Programación</td>
        </tr>
        <tr>
            <td>09:10</td>
            <td>FOL</td>
            <td>Bases de datos</td>
            <td>FOL</td>
            <td>Programación</td>
            <td>Programación</td>
        </tr>
        <tr>
            <td>10:00</td>
            <td>Bases de datos</td>
            <td>Lenguajes de marcas</td>
            <td>FOL</td>
            <td>Programación</td>
            <td>Sistemas operativos</td>
        </tr>
    </table>
```

A las tablas se les puede poner un título o texto explicativo usando la etiqueta `<caption>`. Los navegadores dan a este título el mismo ancho que a la tabla, por lo que cuando es muy larga ocupa más de una línea.

Combinar celdas

Se pueden combinar celdas usando los atributos `colspan` y `rowspan` de las etiquetas `<th>` y `<td>`.

- `Rowspan`, sirve para indicar el número de filas que ocupará una celda. Por ejemplo: `<td rowspan=2></td>` serviría para crear una celda que visualmente sería el resultado de unir verticalmente dos celdas de la misma columna.
- `Colspan`, se usa para unir celdas horizontalmente. Así `<td colspan=3></td>` produciría una celda que se extendería horizontalmente a lo largo de 3 columnas.

Como se ve en el ejemplo siguiente:

<table>
<tr><td><TD>...</TD></td><td><TD>...</TD></td><td><TD>...</TD></td><td rowspan="3"><TD rowspan=”3”>...</TD></td></tr>
<tr><td colspan="2"><TD colspan=”2”>...</TD></td><td><TD>...</TD></td></tr>
<tr><td><TD>...</TD></td><td><TD>...</TD></td><td><TD>...</TD></td></tr>
</table>

Tablas dentro de otras tablas

Para generar tablas más complejas se pueden meter elementos `<table>` dentro de otras tablas. Para ello hay que insertar una etiqueta `<table>` (con todos sus elementos de fila y columna) dentro de un elemento `<td>` o `<th>`. Por ejemplo:

```
<table>
    <tr>
        <td>F1_C1</td>
        <td>F1_C2</td>
        <td>
            <table>
               <tr>
                    <td>F1_C3 F1_C1</td>
                    <td>F1_C3 F1_C2</td>
                </tr>
                <tr>
                    <td>F1_C3 F2_C1</td>
                    <td>F1_C3 F2_C2</td>
                </tr>
            </table>
```

```
            </td>
        </tr>
        <tr>
            <td>
                <table>
                    <tr>
                        <td>F2_C1 F1_C1</td>
                        <td>F2_C1 F1_C2</td>
                        <td>F2_C1 F1_C3</td>
                    </tr>
                </table>
            </td>
            <td>F2_C2</td>
               <td>F2_C3</td>
        </tr>
    </table>
```

<table>
<tr><td colspan="3">F1_C1</td><td>F1_C2</td><td>F1_C3 F1_C1</td><td>F1_C3 F1_C2</td></tr>
<tr><td></td><td></td><td></td><td></td><td>F1_C3 F2_C1</td><td>F1_C3 F2_C2</td></tr>
<tr><td>F2_C1 F1_C1</td><td>F2_C1 F1_C2</td><td>F2_C1 F1_C3</td><td>F2_C2</td><td colspan="2">F2_C3</td></tr>
</table>

1.4.2 Tablas en HTML5

Con HTML5 se han incorporado nuevas etiquetas que dotan de mayor semántica a las tablas y permiten controlar mejor su presentación con la ayuda de CSS.

Hml 5 dispone de tres etiquetas que nos ayudan a organizar mejor las tablas: `<thead>`, `<tbody>`, `<tfoot>`. Con estas etiquetas podemos agrupar filas de una tabla, creando "agrupaciones" a las que después podemos aplicar un estilo determinado con las hojas de estilo de una sola vez (sin necesidad de indicarlo fila a fila o celda a celda).

- `<thead>` representa la *cabecera* de la tabla
- `<tbody>` representa el *cuerpo* de la tabla
- `<tfoot>` representa el *pie* de la tabla

Una tabla puede tener uno o varios `<tbody>`, pero tanto la cabecera de la tabla (`<thead>`) como el pie de tabla (`<tfoot>`) son opcionales y sólo puede haber uno de cada en una tabla.

A continuación vemos un ejemplo del uso de estas etiquetas para distribuir la tabla que se muestra en la imagen:

Ventas anuales

Análisis de ventas anuales

AÑO	Expansión de ventas		
	Producto A	Producto B	Producto C
2013	1010	2332	4444
2012	330	401	3223
2011	110	10	69
Total	1450	6333	7736

```
<table>
 <caption>
    Análisis de ventas anuales
 </caption>

 <thead>
   <tr>
     <th rowspan="2" scope="col">AÑO</th>
     <th colspan="3" scope="col">Expansión de ventas</th>
   </tr>
   <tr>
     <th>Producto A</th><th>Producto B</th><th>Producto C</th>
   </tr>
 </thead>

 <tfoot>
   <tr>
```

```
      <th>Total</th><th>1450</th><th>6333</th><th>7736</th>
    </tr>
  </tfoot>

  <tbody>
    <tr>
      <th scope="row">2013</th><td>1010</td><td>2332</td><td>4444</td>
    </tr>
    <tr>
      <th scope="row">2012</th><td>330</td><td>401</td><td>3223</td>
    </tr>
    <tr>
      <th scope="row">2011</th><td>110</td><td>10</td><td>69</td>
    </tr>
  </tbody>
</table>
```

Además, las tablas en Html 5 disponen de dos etiquetas que podemos usar para indicar con precision que columnas queremos resaltar por medio de las hojas de estilo. Estas directivas son: `<colgroup> y <col>`.

La etiqueta vacía `<col>` permite hacer referencia a una columna y la etiqueta `<colgroup>` permite definir grupos de columnas (de manera similar a como la etiqueta `<tbody>` define grupos de filas). Esta etiqueta debe de ser la primera de la tabla excepto si se usa título en la tabla (`<caption>`) en cuyo caso será la segunda.

Cada `<col />` hace referencia a una columna. Por ejemplo, si añadimos la siguiente línea al ejemplo anterior a continuación de la etiqueta table:

```
<col style="width:40%" /><col style=" width:32%" />
<col style=" width:14%" /><col style=" width:23%" />
```

Estaríamos haciendo referencia a cada una de las columnas con la etiqueta `col`, con lo cual la tabla se visualizaría:

Análisis de ventas anuales

AÑO	Expansión de ventas		
	Producto A	Producto B	Producto C
2013	1010	2332	4444
2012	330	401	3223
2011	110	10	69
Total	1450	6333	7736

La etiqueta `<colgroup>` permite agrupar varias columnas de la tabla. Para indicar el número de columnas que abarca la agrupación se usa el atributo `span`. Vamos a ver un ejemplo de agrupación de columnas realizada con la etiqueta `<colgroup>`: Si añadimos a nuestra tabla inicial las siguientes líneas después del `<caption>`

```
<colgroup span="1" style="width:100px;border:3px solid navy;"/>
<colgroup span="3" style=" width:150px;" />
```

Análisis de ventas anuales

AÑO	Expansión de ventas		
	Producto A	Producto B	Producto C
2013	1010	2332	4444
2012	330	401	3223
2011	110	10	69
Total	1450	6333	7736

Estaremos creando dos grupos de columnas, uno de una única columna que tendrá de ancho 100px y borde azul, y otro de tres columnas que tendrán de ancho 150px como se ve en la imagen.

Las propiedades aplicables son sólo 4 y en casos muy concretos: `border`, `background`, `width` y `visibility`.

La etiqueta `<colgroup>` puede usarse también como un simple contenedor que agrupa todas las directiva `<col>`. En este caso, el atributo `span` debe utilizarse únicamente con la directiva `<col>`. Así, podríamos haber obtenido el mismo resultado si ponemos:

```
<colgroup>
  <col span="1" style="width:100px;border:3px solid navy;" />
  <col span="3" style="width:150px;" />
</colgroup>
```

Accesibilidad en las tablas

Cuando una persona ciega se encuentra una tabla mientras navega con los navegadores especiales de voz o traductores a braile, es muy difícil que se entere como esta estructurada la informacion . Para estas personas existen una serie de atributos Html que les facilitan la tarea de comprender como esta estructurada la informacion de una tabla.

`scope` y `headers` son dos atributos de las etiquetas `<th>` y `<td>`, pensados para personas con discapacidades visuales (ciegas), que usan navegadores no visuales.

scope

Este atributo especifica el conjunto de celdas de datos para las cuales la celda de encabezado actual proporciona información de encabezado. Este atributo puede usarse en lugar del atributo `headers`, en particular en tablas sencillas. Si se especifica, este atributo debe tener uno de los siguientes valores:

- `row` (fila): La celda actual proporciona información de encabezado para el resto de la fila que la contiene
- `col` (columna): La celda actual proporciona información de encabezado para el resto de la columna que la contiene
- `rowgroup` (grupo de filas): La celda de encabezado proporciona información de encabezado para el resto del grupo de filas que la contiene.
- `colgroup` (grupo de columnas): La celda de encabezado proporciona información de encabezado para el resto del grupo de columnas que la contiene.

En el siguiente ejemplo vemos el uso de este atributo:

Lista de precios

	Frutas (scope="colgroup")		
Cantidad (scope="col")	Manzanas (scope="col")	Naranjas (scope="col")	Bananas (scope="col")
1 Kilo (scope="row")	2 €	4 €	6 €
2 Kilos (scope="row")	4 €	6 €	8 €

```
<table>
   <caption>
        Lista de precios
   </caption>
   <tr>
     <th></th>
     <th colspan="3" scope="colgroup">Frutas <span class="rango">
                                        (scope="colgroup") </span> </th>
   </tr>
   <tr>
     <th scope="col">Cantidad <em>(scope="col")</em></th>
     <th scope="col">Manzanas <em>(scope="col")</em></th>
     <th scope="col">Naranjas <em>(scope="col")</em></th>
     <th scope="col">Bananas <em>(scope="col")</em></th>
    </tr>
    <tr>
     <th scope="row">1 Kilo <em>(scope="row")</em></th>
     <td>2 €</td>
     <td>4 €</td>
     <td>6 €</td>
    </tr>
    <tr>
     <th scope="row">2 Kilos <em>(scope="row")</em></th>
```

```
    <td>4 €</td>
    <td>6 €</td>
    <td>8 €</td>
  </tr>
</table>
```

`headers`

Una alternativa a usar el atributo scope es usar el atributo `id` y `headers` para crear asociaciones entre cabeceras y celdas. Para hacerlo tenemos que:

1. Añadir un atributo `id` a las celdas de encabezado (`<th>`)
2. Añadir el atributo `headers` a cada celda de datos (`<td>`). Este atributo contiene la lista de `id` de celdas de encabezado que proporcionan información de encabezado para la celda de datos actual. El valor de este atributo es la lista de ids separados por espacios.

   Vamos a ver el uso de este atributo con un ejemplo:

```
<table>
 <caption>Notas de los alumnos</caption>
 <tr>
    <th id="t1">Nombre <em>(id="t1")</em></th>
    <th id="t2">Comportamiento <em>(id="t2")</em></th>
    <th id="t3">Trabajos <em>(id="t3")</em></th>
    <th id="t4">Examen <em>(id="t4")</em></th>
 </tr>
 <tr>
    <td id="nombre" headers="t1">Lucia <em>(headers="t1")</em></td>
    <td headers="nombre t2">Muy bueno <em>(headers="nombre t2")</em>
    </td>
    <td headers="nombre t3">8 <em>(headers="nombre t3")</em></td>
    <td headers="nombre t4">7 <em>(headers="nombre t4")</em></td>
 </tr>
 <tr>
   <td id="nombre2" headers="t1">Jorge <em>(headers="t1")</em></td>
   <td headers="nombre2 t2">Bueno <em>(headers="nombre2 t2")</em></td>
   <td headers="nombre2 t3">8 <em>(headers="nombre2 t3")</em></td>
   <td headers="nombre2 t4">6 <em>(headers="nombre2 t4")</em></td>
 </tr>
</table>
```

Notas de los alumnos

Nombre *(id="t1")*	Comportamiento *(id="t2")*	Trabajos *(id="t3")*	Examen *(id="t4")*
Lucia *(headers="t1")*	Muy bueno *(headers="nombre t2")*	8 *(headers="nombre t3")*	7 *(headers="nombre t4")*
Jorge *(headers="t1")*	Bueno *(headers="nombre t2")*	8 *(headers="nombre t3")*	6 *(headers="nombre t4")*

1.4.3 Dar formato a tablas con CSS

La mayoría de propiedades de uso habitual en CSS (bordes, colores, tipos de letra, ...) son aplicables a las tablas. Pero además, existen propiedades CSS que, explícitamente, sirven para modificar la forma de visualizar los elementos de una tabla.

Espaciado y borde

Una tabla sin CSS parece vacía (por defecto cuando creamos una tabla no se visualiza ningún borde). Y cuando le ponemos un borde vemos que tenemos un modelo de "bordes independientes". Esto es, cada celda tiene un borde y podemos establecer la distancia entre los bordes de las distintas celdas.

Por lo general, nos gustaría que sólo se viese un borde entre dos celdas. Para establecer

el modo de presentación de los bordes el elemento `table` tiene la propiedad `border-collapse` que puede tomar dos valores:

- `separate`: los bordes se mantienen separados. Es el valor por defecto.

	Lunes	Miércoles	Viernes
16:10	Inglés		Dibujo
18:00	Baloncesto	Tenis	Baloncesto

- `Collapse`: los bordes se unen para formar un único borde, con lo cual no hay espacio entre los bordes.

	Lunes	Miércoles	Viernes
16:10	Inglés		Dibujo
18:00	Baloncesto	Tenis	Baloncesto

Cuando visualizamos los bordes separados podemos usar la propiedad `border-spacing` para indicar el espacio entre los bordes. Se puede indicar una medida, que se aplicará todo alrededor, o dos, donde la primera hará referencia a la distancia horizontal y la segunda a la vertical.

```
border-spacing:4px 7px;
```

Con bordes separados, también disponemos de la propiedad `empty-cells` para indicar si queremos que se visualice o no el borde para las celdas vacías. Por ejemplo con

```
empty-cells: hide;
```

se visualizaría →

	Lunes	Miércoles	Viernes
16:10	Inglés		Dibujo
18:00	Baloncesto	Tenis	Baloncesto

Su valor por defecto es show.

Posición del título

La propiedad `caption-side` permite elegir la posición de la leyenda (`<caption>`) con respecto a la tabla. Los dos valores permitidos son `top` (arriba) y `bottom` (abajo). El valor por defecto es `top`.

Disposición de la tabla

La propiedad `table-layout` permite indicar la forma en la que se calcula la anchura de las celdas de la tabla. Puede tomar los siguientes valores:

- `auto`: Valor por defecto. Aunque hayamos indicado anchura para tabla y/o celdas, su tamaño se determina por el contenido de las mismas.
- `fixed`: el tamaño de las tablas y de las celdas depende de los valores establecidos en las propiedades `width` y `height`.

Realiza las tareas 2 y 3.

1.5 Insertar audio y vídeo

Una de las características más sonadas de HTML 5 es su soporte multimedia, que permite la reproducción de archivos de audio/vídeo, o bien la conexión y reproducción de una fuente de audio/vídeo. Esto se hace en HTML 5 con las etiquetas `audio` y `video`, respectivamente.

Html 5 permite la reproducción de vídeo sin utilizar flash. Los formatos más utilizados

para la reproducción de vídeo en html5 son: .mp4, .ogg y .webm

Para visualizar un vídeo en HTML5 el código es el siguiente:

```
<video src="movie.ogg" controls="controls">
    Su navegador no soporta la etiqueta video de HTML5
</video>
```

El atributo `controls` muestra los botones de play, pause y volumen. Naturalmente, y al igual que ocurre con los campos de formulario, los navegadores muestran los controles de manera diferente, ya que la especificación no indica qué aspecto deben tener. De todas maneras, independientemente del aspecto, todos ellos coinciden en los controles a mostrar: reproducir/pausa, una barra de progreso y un control de volumen.

Se recomienda incluír los atributos `width` y `height` para indicar el tamaño del vídeo en pixels. Si no se indican estas medidas, el navegador utiliza las medidas definidas en el vídeo de origen, si están disponibles. De lo contrario, utiliza las medidas definidas en el fotograma poster, si están disponibles. Si ninguna de estas medidas está disponible, el ancho por defecto es de 300 pixels.

Si únicamente se especifica una de las dos medidas, el navegador automáticamente ajusta la medida de la dimensión no proporcionada, conservando la proporción del vídeo.

Con el atributo `poster` podemos indicar la imagen que el navegador debe mostrar mientras el vídeo se está descargando o hasta que el usuario reproduce el video. Si no se indica se verá el primer fotograma del video.

Entre las etiquetas `<video> </video>` introduciremos un mensaje que se visualizará en los navegadores que no soporten el elemento `video` de HTML5.

audio

El audio se inserta casi de la misma manera que el vídeo:

```
<audio src="archivo.mp3" controls="controls">
    Su navegador no soporta la etiqueta audio de HTML5
</audio>
```

Hay que tener cuidado aquí, ya que no todos los formatos de audio son soportados por todos los navegadores.

Los formatos más utilizados para la reproducción de vídeo en html5 son: .mp3 y .ogg

Otros atributos, aplicables tanto al **audio** como al **video** son:

- `autoplay`: que reproduce automáticamente el audio/video cada vez que se carga la página. No se suele utilizar porque se considera una práctica intrusiva.
- `preload`: permite indicar al navegador que comience la descarga del vídeo antes de que el usuario inicie su reproducción. Sus posibles valores son:
  - `auto` → Indica al navegador que comience la descarga
  - `none` → Indica al navegador que no comience la descarga hasta que lo indique el usuario
  - `metadata` → Carga los metadatos (dimensiones, duración, fotogramas, …) pero espera a desgargar el resto hasta que el usuario lo indique.
- `loop`: que ejecuta la reproducción en bucle (cuando finaliza vuelve a empezar)
- `muted`: Permite que el audio o vídeo se reproduzca inicialmente sin sonido.
- `type`: que indica el tipo de archivo o fuente (audio/mp3; video/mp4; video/ogg...)

1.5.1 Audio y vídeo con varias versiones de un archivo

El elemento `source` permite indicar diferentes versiones de un archivo en los elementos de medios (`audio` y `video`) y las imágenes (`picture`). Cada elemento `source` proporciona una versión diferente del medio (codificado utilizando un códec diferente) o imagen (en diferentes tamaños o formatos). Queda en manos del navegador elegir qué recurso cargar, una decisión que tomará en base a los formatos que soporta y a las dimensiones de la imagen (mejor coincidencia). Ejemplo:

```
<video controls="controls"  poster="fotoVideoXX.jpg" >
    <source  src="videoXX.mp4"  type="video/mp4" />
    <source  src="videoXX.ogv"  type="video/ogg" />
    <p>El navegador no puede mostrar el vídeo. </p>
</video>
```

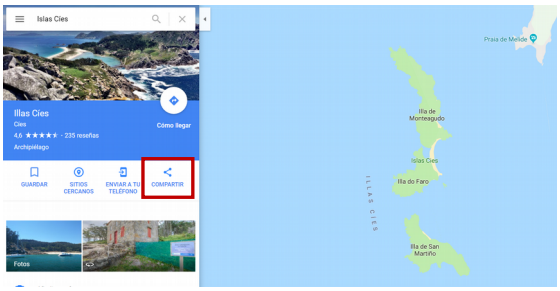
Es importante tener en cuenta que si se declara el atributo `src` en el elemento `audio` o `video`, no será posible indicar alternativas con el elemento `source`.

1.6 Insertar contenido externo

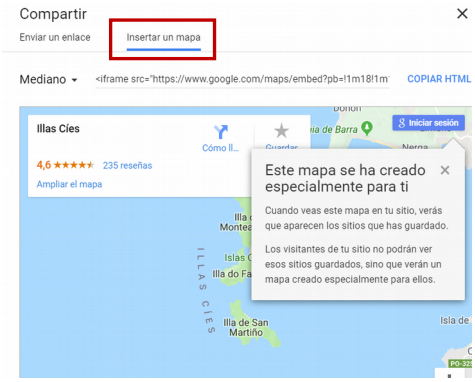
El proceso de insertar contenido de terceros suele ser bastante sencillo y generalmente está explicado en la página de cada red social o plugin.

Vamos a ver, por ejemplo, como insertar un mapa en nuestra página:

1. Seleccionamos la ubicación que queremos y le damos a Compartir



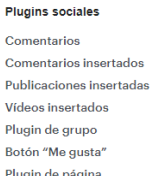
2. Hacemos clic sobre la opción "Insertar un mapa".



3. El sitio nos proporciona el código de una etiqueta `<iframe>` que debemos copiar y pegar en el lugar correspondiente. En esta ventana se nos proporcionan medidas predeterminadas que por lo general no se adaptan a nuestras necesidades, asi que es conveniente que le borremos los atributos que trae por defecto (`width`, `height`, `frameborder` y `style`) y le agreguemos un `class` para darle mediante css, las características que mejor se adapten a nuestro diseño.

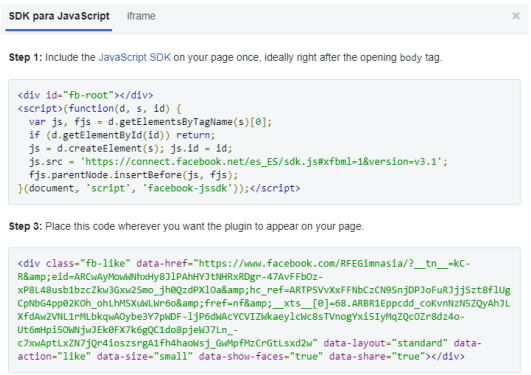
La etiqueta `iframe` se usa para insertar otro documento en nuestro sitio web. Su atributo principal es src que contiene la url de la página incrustada.

Ahora vamos a ver como insertar alguno de los plugins o botones de Facebook:



1. En la página https://developers.facebook.com/docs/plugins aparecen todas las funcionalidades que podemos insertar como el botón "Me gusta" .

2. Seleccionamos por ejemplo este botón y en la página vamos a configurarlo. Seguimos los pasos que allí nos indica y una vez copiamos el código le damos al botón de "Obtener código". Nos aparece el código que debemos copiar en nuestra página para insertar el botón a través de JavaScript o de un `iframe`. Seleccionamos la primera opción que es la recomendada para HTML5.



3. Son dos bloques de código que debemos copiar y pegar. El primero lo vamos a pegar debajo de la etiqueta de apertura del `<body>`, y el segundo bloque de código lo vamos a insertar en el lugar donde queremos que se vea el plugin. Para ver el resultado es necesario subir el sitio al servidor.

1.7 Almacenar datos en local gracias a Web Storage

Cuando queremos almacenar datos de una página Web, por lo general utilizamos cookies. Gracias al nuevo estándar de almacenamiento Web de HTML5 *Web Storage*, podemos guardar más datos que antes. Con este sistema, la información se almacena en una base de datos en el interior de los navegadores.

- Local Storage → Guarda información que permanecerá almacenada por tiempo indefinido; sin importar que el navegador se cierre.
- Session Storage → Almacena los datos de una sesión y éstos se eliminan cuando el navegador se cierra.

  Las características de Local Storage y Session Storage son:
  - Permiten almacenar entre 5MB y 10MB de información; incluyendo texto y multimedia
  - La información está almacenada en la computadora del cliente y NO es enviada en cada petición del servidor, a diferencia de las cookies
  - Utilizan un número mínimo de peticiones al servidor para reducir el tráfico de la red
  - Previenen pérdidas de información cuando se desconecta de la red
  - La información es guardada por domino web (incluye todas las páginas del dominio)

Para almacenar datos de una página Web es indispensable utilizar JavaScript.

1.8 Otras características de HTML5

- **Canvas**: El elemento canvas puede definirse como un entorno para crear imágenes dinámicas. El propio elemento es relativamente simple. Lo único que hay que especificar al usarlo son sus dimensiones:

```
<canvas id="entorno_canvas" width="360" height="240">;
</canvas>
```

Cualquier cosa que escribamos entre la apertura y cierre de la etiqueta canvas solamente será interpretado por navegadores que no soportan aún la nueva etiqueta.

```
<canvas id="entorno_canvas" width="360" height="240">
    <p>Tu navegador no soporta canvas: Aquí deberías de ver una imagen :</p>
    <img src="html5_logo.png" alt="HTML5" />
</canvas>
```

En el ejemplo anterior, la imagen solo sería visible por aquellos navegadores que aún no soporten canvas.

Canvas es un nuevo componente que permitirá dibujar, por medio de las funciones de un API, en la página todo tipo de formas, que podrán estar animadas y responder a interacción del usuario. El potencial de canvas reside en su habilidad para actualizar su contenido en tiempo real. Si usamos esa habilidad para responder a eventos de usuario, podemos crear herramientas y juegos que anteriormente a la nueva especificación hubiesen requerido de un plugin externo como Flash.

- **Web Workers**: son procesos que requieren bastante tiempo de procesamiento por parte del navegador, pero que se podrán realizar en un segundo plano, para que el usuario no tenga que esperar que se terminen para empezar a usar la página. Para ello se dispondrá también de un API para el trabajo con los Web Workers.
- **Geolocalización**: La API de geolocalización permite al usuario compartir su ubicación a las aplicaciones web si así lo desea. Por razones de privacidad, al usuario se le pide que confirme el permiso para proporcionar información de ubicación.


Realiza las tareas 4 y 5.