



Desarrollo de aplicaciones multiplataforma

Elaborado por: José Antonio Sánchez

Módulo 3

Programación HTML5

SESIÓN 2

4.1. Uso de las etiquetas semánticas

`<header></header>`

“Un grupo de navegación o contenido introductorio. Un elemento header normalmente contiene una sección de encabezado (un elemento h1-h6 o un elemento hgroup), pero puede contener otro tipo de elementos, como una tabla de contenidos, un formulario de búsqueda o cualquier logo importante.”

4.1. Uso de las etiquetas semánticas

```
<header></header>
```

```
<div id="header">
```

```
  <a href="/"><img src=logo.png alt="home"></a>
```

```
  <h1>My Weblog</h1>
```

```
  <p class="tagline">
```

```
    A lot of effort went into making this effortless.
```

```
  </p>
```

```
</div>
```

4.1. Uso de las etiquetas semánticas

```
<header></header>
```

```
<header>
```

```
  <a href="/"><img src=logo.png alt="home"></a>
```

```
  <hgroup>
```

```
    <h1>Title</h1>
```

```
    <h2 class="tagline">
```

```
      A lot of effort went into making this effortless.
```

```
    </h2>
```

```
  </hgroup>
```

```
</header>
```

* Se utiliza <hgroup> para indicar la relación entre el <h1> y el <h2>

4.1. Uso de las etiquetas semánticas

`<nav></nav>`

“El elemento `<nav>` representa una sección de una página que enlaza con otras páginas o partes de la misma página: una sección con enlaces de navegación.”

No estamos obligados a utilizar un único elemento `<nav>` en toda la página: navegación principal en la cabecera, una tabla de contenidos, enlaces en el pie de la página...

No todos los enlaces estarán dentro de etiquetas `<nav>`

4.1. Uso de las etiquetas semánticas

```
<nav></nav>
```

```
<div id="nav">
```

```
  <ul>
```

```
    <li><a href="#">home</a></li>
```

```
    <li><a href="#">blog</a></li>
```

```
    <li><a href="#">gallery</a></li>
```

```
    <li><a href="#">about</a></li>
```

```
  </ul>
```

```
</div>
```


4.1. Uso de las etiquetas semánticas

```
<nav></nav>
```

```
<nav>
```

```
  <ul>
```

```
    <li><a href="#">home</a></li>
```

```
    <li><a href="#">blog</a></li>
```

```
    <li><a href="#">gallery</a></li>
```

```
    <li><a href="#">about</a></li>
```

```
  </ul>
```

```
</nav>
```

4.1. Uso de las etiquetas semánticas

`<footer> </footer>`

“Representa el pie de una sección. Un pie tradicionalmente contiene información acerca de su sección, como quién escribió el contenido, enlaces relacionados, copyright y similares.”

Podemos tener tantos elementos `<footer>` como sea necesario aunque al menos debería tener un pie principal: privacidad, condiciones del servicio, copyright, mapa del sitio web, accesibilidad, contacto y otros muchos enlaces que pueden ir incluidos en un elemento `<nav>`

4.1. Uso de las etiquetas semánticas

```
<footer> </footer>
```

```
<div id="footer">  
  <p>#167;</p>  
  <p>#169; 2013#8211;9 <a href="#">Arkaitz Garro</a></p>  
</div>
```

```
<footer>  
  <p>#167;</p>  
  <p>#169; 2013#8211;9 <a href="#">Arkaitz Garro</a></p>  
</footer>
```

4.1. Uso de las etiquetas semánticas

```
<aside></aside>
```

```
<div id="footer">
```

```
  <p>#167;</p>
```

```
  <p>#169; 2013#8211;9 <a href="#">Arkaitz Garro</a></p>
```

```
</div>
```

```
<footer>
```

```
  <p>#167;</p>
```

```
  <p>#169; 2013#8211;9 <a href="#">Arkaitz Garro</a></p>
```

```
</footer>
```

4.1. Uso de las etiquetas semánticas

```
<aside></aside>
```

```
<div class="aside">  
  <div class="related"></div>  
  <div class="related"></div>  
  <div class="related"></div>  
</div>
```

```
<aside>  
  <div class="related"></div>  
  <div class="related"></div>  
  <div class="related"></div>  
</aside>
```

4.1. Uso de las etiquetas semánticas

`<figure></figure>`

“El elemento figure representa algún “contenido corriente” (“flow content”), opcionalmente con una leyenda, que está contenida en sí mismo y está típicamente referenciado como una unidad del flujo principal del documento.

Por lo tanto el elemento puede ser usado para ilustraciones, diagramas, fotos, listas de código, etc, que están referidos desde el contenido principal del documento, pero eso puede ser movido, sin afectar el flujo del documento, fuera del contenido primario, por ejemplo al lateral de una página, a páginas dedicadas, o a un apéndice.”

4.1. Uso de las etiquetas semánticas

`<figure></figure>`

`<figure>`

``

`<figcaption>`

Bruce and Remy welcome questions

`<small>Photo © Bruce's mum</small>`

`</figcaption>`

`</figure>`

“alt” indica el texto a mostrar cuando la imagen no está disponible. No es sustituible por figcaption

4.1. Uso de las etiquetas semánticas

`<article></article>`

“Este elemento representa un contenido completo, auto-contenido en un documento, página, aplicación o sitio web, que es, en principio, independiente de ser distribuido y reutilizado, por ejemplo en un RSS. Puede ser un post de un foro, un artículo de un periódico o revista, una entrada de un blog, un comentario de un usuario, un widget o cualquier otro elemento independiente.

Cuando los artículos están anidados, los artículos interiores representan contenido que en principio está relacionado con el artículo que los contiene. Por ejemplo, una entrada de un blog puede aceptar comentarios de usuarios, que están incluidos dentro del contenido principal y relacionados con el mismo.”

4.1. Uso de las etiquetas semánticas

```
<article></article>
```

```
<div class="article">  
  <p class="post-date">October 22, 2009</p>  
  <h2>  
    <a href="#" title="link to this post">Travel day</a>  
  </h2>  
  <div class="content">  
    Content goes here...  
  </div>  
  <div class="comments">  
    <p><a href="#">3 comments</a></p>  
  </div>  
</div>
```

4.1. Uso de las etiquetas semánticas

```
<article></article>
```

```
<article>
  <time datetime="2009-10-22">October 22, 2009</time>
  <h2>
    <a href="#" title="link to this post">Travel day</a>
  </h2>
  <div class="content">
    Content goes here...
  </div>
  <div class="comments">
    <p><a href="#">3 comments</a></p>
  </div>
</article>
```

4.1. Uso de las etiquetas semánticas

`<section></section>`

“Representa una sección genérica de un documento o aplicación. Una sección, en este contexto, es un grupo temático de contenido, que generalmente incluye una cabecera.”

Este elemento es utilizado para dividir el documento o un artículo en diferentes áreas

4.1. Uso de las etiquetas semánticas

```
<section></section>
```

```
<div class="articles">  
  <div class="article">  
    <p class="post-date">October 22, 2009</p>  
    <h2>  
      <a href="#" title="link to this post">Travel day</a>  
    </h2>  
    <div class="content">  
      Content goes here...  
    </div>  
    <div class="comments">  
      <p><a href="#">3 comments</a></p>  
    </div>  
  </div>  
</div>
```

4.1. Uso de las etiquetas semánticas

```
<section></section>
```

```
<section class="articles">
```

```
  <article>
```

```
    <time datetime="2009-10-22">October 22, 2009</time>
```

```
    <h2>
```

```
      <a href="#" title="link to this post">Travel day</a>
```

```
    </h2>
```

```
    <div class="content">
```

```
      Content goes here...
```

```
    </div>
```

```
    <section class="comments">
```

```
      <p><a href="#">3 comments</a></p>
```

```
    </section>
```

```
  </article>
```

```
</section>
```

4.1. Uso de las etiquetas semánticas

<section></section>

<h1>Rules for Munchkins</h1>

<h2>Yellow Brick Road</h2>

<p>It is vital that Dorothy follows it—so no selling bricks as "souvenirs"</p>

<h2>Fan Club uniforms</h2>

<p>All Munchkins are obliged to wear their "I'm a friend of Dorothy!" t-shirt when representing the club</p>

<p>Vital caveat about the information above: does not apply on the first Thursday of the month.</p>

¿¿ El texto "Vital caveat..." hace referencia a "Fan Club uniforms" o a todo el documento ??

4.1. Uso de las etiquetas semánticas

<section></section>

<article>

<h1>Rules for Munchkins</h1>

<section>

<h2>Yellow Brick Road</h2>

<p>It is vital that Dorothy follows it—so no selling bricks as "souvenirs"</p>

</section>

<section>

<h2>Fan Club uniforms</h2>

<p>All Munchkins are obliged to wear their "I'm a friend
of Dorothy!" t-shirt when representing the club</p>

</section>

<p>Vital caveat about the information above:

does not apply on the first Thursday of the month.</p>

</article>

4.1. Uso de las etiquetas semánticas

```
<section></section>
```

```
<article>
```

```
  <h1>Rules for Munchkins</h1>
```

```
  <section>
```

```
    <h2>Yellow Brick Road</h2>
```

```
    <p>It is vital that Dorothy follows it—so no selling bricks as "souvenirs"</p>
```

```
  </section>
```

```
  <section>
```

```
    <h2>Fan Club uniforms</h2>
```

```
    <p>All Munchkins are obliged to wear their "I'm a friend  
      of Dorothy!" t-shirt when representing the club</p>
```

```
    <p><strong>Vital caveat about the information above:  
      does not apply on the first Thursday of the month.</strong></p>
```

```
  </section>
```

```
</article>
```


4.1. Uso de las etiquetas semánticas

<article> y <section>

Entonces una “section” es una parte de un “article”, ¿no?

Además de crear secciones dentro de un artículo para aportar significado al mismo, podemos crear secciones para tener categorías de artículos.

4.1. Uso de las etiquetas semánticas

<article> y <section>

<section>

<h1>Articles about llamas</h1>

<article>

<h2>The daily llama: Buddhism and South American camelids</h2>

<p>blah blah</p>

</article>

<article>

<h2>Shh! Do not alarm a llama</h2>

<p>blah blah</p>

</article>

</section>

4.1. Uso de las etiquetas semánticas

<article> y <section>

<section>

<h1>Articles about root vegetables</h1>

<article>

<h2>Carrots: the orange miracle</h2>

<p>blah blah</p>

</article>

<article>

<h2>Eat more Swedes (the vegetables, not the people)</h2>

<p>blah blah</p>

</article>

</section>

4.3. Otras etiquetas semánticas

http://www.w3schools.com/html/html5_semantic_elements.asp

Tag	Description
<u><article></u>	Defines an article
<u><aside></u>	Defines content aside from the page content
<u><details></u>	Defines additional details that the user can view or hide
<u><figcaption></u>	Defines a caption for a <figure> element
<u><figure></u>	Specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.
<u><footer></u>	Defines a footer for a document or section
<u><header></u>	Specifies a header for a document or section
<u><main></u>	Specifies the main content of a document
<u><mark></u>	Defines marked/highlighted text
<u><nav></u>	Defines navigation links
<u><section></u>	Defines a section in a document
<u><summary></u>	Defines a visible heading for a <details> element
<u><time></u>	Defines a date/time

Nuevos elementos en HTML5:

http://www.w3schools.com/html/html5_new_elements.asp

4.4. Ejercicio práctico

<http://www.arkaitzgarro.com/html5/capitulo-18.html#ej01>

Dada la siguiente página web, estructurada en XHTML, aplicar las nuevas etiquetas semánticas de HTML5 donde sea conveniente, manteniendo el mismo aspecto visual (modificar la hoja de estilos si es necesario) y funcionalidad. Realizad también los cambios necesarios en la cabecera del documento (elemento <head>).

4.5. Atributos globales

accesskey

Permite especificar un atajo de teclado para activar un elemento y asignarle el foco

Este atributo ya existía en HTML 4, pero ahora está disponible para cualquier elemento

Para evitar conflictos con otros atajos, ahora se permite asignar alternativas:

```
<input type="search" name="q" accesskey="s 0">
```

Este elemento es accesible a través de dos atajos de teclado: la tecla “s” o la tecla “0”, por ese orden

4.5. Atributos globales

contenteditable

Inventado por Microsoft, pero ahora parte de la especificación oficial

- ✓ Los usuarios pueden editar los contenidos de un elemento que incluya esta etiqueta
- ✓ Este elemento debe ser seleccionable y el navegador debe proporcionar una marca que indique la posición actual del cursor
- Es posible cambiar el formato del texto del contenido, añadiendo negritas, cambiar la fuente, añadir listas, etc.

4.5. Atributos globales

contenteditable

Este atributo es de tipo booleano, por lo que su valor puede ser true o false

Para utilizarla desde JavaScript la notación es “contentEditable”, además, existe otra propiedad llamada isContentEditable

El contenido que ha sido seleccionado por el usuario, puede ser objeto de modificaciones mediante el comando *element.execCommand()* indicando el tipo de modificación: poner en negrita, copiar, cambiar la fuente...

```
document.execCommand('bold',false,null);
```


4.5. Atributos globales

contenteditable

Es requisito que el documento se haya marcado como editable:

```
document.designMode = 'on';
```

Si se desea almacenar los cambios realizados, es necesario enviarlo al servidor

Como no existe ningún API que nos posibilite esta acción, debemos utilizar algún tipo de tecnología tipo AJAX

4.5. Atributos globales

contenteditable

Ejercicio práctico:

Crear un documento que tenga una sección editable y mediante botones permita realizar cambios en el texto: negrita, ...

4.5. Atributos globales

*Custom data attributes. data-**

HTML5 permite crear atributos personalizados para los elementos

Estos atributos son utilizados para pasar información a JavaScript

```
<ul id="vegetable-seeds">  
  <li data-spacing="10cm" data-sowing-time="March to June">Carrots</li>  
  <li data-spacing="30cm" data-sowing-time="February to March">Celery</li>  
  <li data-spacing="3cm" data-sowing-time="March to September">Radishes</li>  
</ul>
```

4.5. Atributos globales

draggable

Este atributo indica que el elemento indicado puede ser *arrastrable*

Más adelante trabajaremos con él

5. Elementos de formulario

HTML5 hace que el desarrollo de formularios sea mucho más sencillo:

- ✓ Se han añadido dos nuevos métodos que pueden ser utilizados en la acción del formulario: update y delete
- ✓ Existen nuevos tipos de input y elementos de formulario que mejoran la experiencia del usuario y facilitan el desarrollo de los formularios
- ✓ En algunos casos proporcionan validaciones propias de sus datos

5.1. Tipos de input

La especificación de HTML5 define 12 nuevos tipos de input que podemos utilizar en nuestros formularios

No se define cómo deben mostrarse los nuevos tipos en los navegadores, ni los campos ni las validaciones

Por esta razón, los navegadores que no comprendan los nuevos tipos de entrada, mostrarán un campo de texto tradicional

ii Compatibilidad con navegadores antiguos está garantizada !!

5.1. Tipos de input

1. <input type="email">

Indica al navegador que no debe permitir que se envíe el formulario si el usuario no ha introducido una dirección de email válida

Sólo comprueba el formato no si la dirección existe o no

Puede enviar este campo vacío a menos que se indique que es obligatorio

Tiene un atributo “multiple” para indicar una lista de emails válidos, separados por comas

5.1. Tipos de input

2. `<input type="url">`

Indica al navegador que no debe permitir que se envíe el formulario si el usuario no ha introducido una URL correcta

Algunos navegadores ofrecen ayudas al usuario(Opera que añade el prefijo `http://`)

Una URL no tiene que ser necesariamente una dirección web, sino que es posible utilizar cualquier formato de URI válido, como por ejemplo `tel:555123456`

5.1. Tipos de input

3. `<input type="date">`

Es de los más esperados y útiles ya que existen muchos formatos de fecha diferentes, y supone un inconveniente para los desarrolladores o los propios usuarios

El navegador el que proporciona la interfaz de usuario para el calendario, e independientemente del formato en el que se muestre, los datos que se envían al servidor cumplen la norma ISO para el formato de fechas

5.1. Tipos de input

4. `<input type="time">`

Permite introducir una hora en formato 24h, y validarlo

El navegador el encargado de mostrar la interfaz de usuario correspondiente: puede ser un simple campo para la hora y los minutos o mostrar algo más complejo como un reloj de agujas

5.1. Tipos de input

5. <input type="datetime">

Este nuevo tipo de campo es la combinación de los tipos date y time, por lo que se valida tanto la fecha como la hora introducida

5.1. Tipos de input

6. `<input type="month">`

Permite la selección de un mes en concreto

La representación interna del mes es un valor entre 1 y 12, pero es el navegador quien decide la interfaz

5.1. Tipos de input

7. `<input type="week">`

Permite la selección de una semana del año concreta

La representación interna del mes es un valor entre 1 y 53, por ejemplo la semana 7 sería: 2014-W07

Queda en manos del navegador la manera de mostrarlo al usuario

5.1. Tipos de input

8. `<input type="number">`

Valida la entrada de un tipo de dato numérico

Permite los atributos min, max y step

5.1. Tipos de input

9. `<input type="range">`

Muestra un control deslizante en el navegador

Para conseguir un elemento de este estilo, era necesario un gran esfuerzo para combinar imágenes, funcionalidad y accesibilidad

Este tipo de campo también permite los atributos min, max y step

5.1. Tipos de input

10. `<input type="tel">`

Espera que se proporcione un número de teléfono

No se realiza ninguna validación ya que un número de teléfono puede representarse de muchas maneras: +44 (0) 208 123 1234

En dispositivos con un teclado virtual, éste se adaptará para mostrar únicamente los caracteres asociados con números de teléfono

5.1. Tipos de input

11. `<input type="search">`

Espera que se proporcione un término de búsqueda

La diferencia con un campo de texto normal, es únicamente estética

Puede ofrecer alguna funcionalidad extra como un histórico

Por norma general, toma el aspecto de un campo de búsqueda del navegador

5.1. Tipos de input

12. `<input type="color">`

Permite seleccionar un color de una paleta de colores mostrada por el navegador

Envía el color con código hexadecimal, por ejemplo el negro: `%23000000`

5.2. Nuevos atributos

list y <datalist>

La combinación del atributo list y un elemento de tipo <datalist> da como resultado un campo de texto, donde el usuario puede introducir cualquier contenido, y las opciones definidas en el <datalist> se muestran como una lista desplegable

Hay que tener en cuenta que la lista tiene que estar contenida en un elemento <datalist> cuyo id coincide con el indicado en el atributo list

5.2. Nuevos atributos

list y <datalist>

```
<input id="form-person-title" type="text" list="mylist">
```

```
<datalist id="mylist">
```

```
  <option label="Mr" value="Mr">
```

```
  <option label="Ms" value="Ms">
```

```
  <option label="Prof" value="Mad Professor">
```

```
</datalist>
```

5.2. Nuevos atributos

autofocus

El atributo booleano autofocus permite definir que control va a tener el foco cuando la página se haya cargado

Hasta ahora, esto se conseguía a través de JavaScript, utilizando el método .focus() en un elemento concreto, al cargarse el documento

Ahora es el navegador el encargado de esta tarea, y puede comportarse de manera más inteligente, como no cambiando el foco de un elemento si el usuario ya se encuentra escribiendo en otro campo

5.2. Nuevos atributos

autofocus

Únicamente debe existir un elemento con este atributo definido en el documento

Debe utilizarse únicamente cuando el control que recibe el foco es el elemento principal de la página, como por ejemplo en un buscador

5.2. Nuevos atributos

placeholder

Se utiliza para mejorar la usabilidad de los formularios

Se trata de un pequeño texto de ayuda dentro del campo input que desaparece cuando el usuario introduce algún dato

Este atributo no debe sustituir a la etiqueta <label>

5.2. Nuevos atributos

required

Puede ser utilizado tanto en un `<textarea>` como en casi todos los elementos `<input>` (excepto `hidden`, `image` o `submit`)

El navegador no permite el envío del formulario si el campo en concreto está vacío (`required = "true"`)

5.2. Nuevos atributos

multiple

Este atributo permite definir que un campo puede admitir varios valores, como URLs o emails

Un uso muy interesante de este atributo es utilizarlo en conjunto con el campo `<input type="file">`, para seleccionar varios ficheros

```
<input type="file" multiple="multiple">
```

5.2. Nuevos atributos

autocomplete

Permite añadir la funcionalidad de auto completado en algunos campos de formulario

Es una característica que lleva mucho tiempo siendo utilizada (desde IE5) pero ha sido incluida ahora en el estándar de HTML5

Este atributo permite controlar el comportamiento del auto completado en los campos de texto del formulario (que por defecto está activado)

5.2. Nuevos atributos

step

El atributo step controla los “pasos” por los que un campo aumenta o disminuye su valor

Por ejemplo, para introducir un porcentaje en múltiplos de 5:

```
<input type="range" min="0" max="100" step="5">
```

5.2. Nuevos atributos

pattern

Permite definir una expresión regular que el valor del campo debe cumplir

Algunos de los tipos de input (email, number, url...), son realmente expresiones regulares

Por ejemplo, si el usuario debe introducir un número seguido de tres letras mayúsculas, podríamos definir esta expresión regular:

```
<input pattern="[0-9][A-Z]{3}" name="part"
  title="A part number is a digit followed by three uppercase letters.">
```

5.2. Nuevos atributos

form

Permite relacionar el campo con el formulario al que pertenece a través del “id”, pudiendo colocar este elemento en cualquier lugar de la página

En HTML4, si por la razón que fuese (principalmente diseño) un elemento tuviese que mostrarse apartado del resto de elementos del formulario, se hacía casi necesario incluir toda la página dentro de una etiqueta <form>

```
<form id="foo">  
  <input type="text">  
  ...  
</form>  
<textarea form="foo"></textarea>
```

5.3. Nuevos elementos

<progress>

Representar un avance o progreso en la ejecución de una tarea

Define los siguientes atributos:

- ✓ max: define el trabajo total, la duración, el valor máximo... Su valor por defecto es 1.0
- ✓ value: el valor actual (en coma flotante). Su valor debe ser mayor o igual a 0.0 y menor que el valor especificado en max (o 1.0)
- ✓ position: atributo de sólo lectura que representa la posición actual del elemento <progress>. Este valor es igual a value/max y -1 si no se puede determinar la posición

Las unidades son arbitrarias, y no se especifican

5.3. Nuevos elementos

`<meter>`

Es muy similar a `<progress>`, de hecho, se discute la necesidad de disponer de dos elementos tan similares

No debe utilizarse para medir progreso (semántica)

Esta nueva etiqueta se usa para representar escalas de medidas conocidas, como la longitud, masa, peso, uso de disco, entre otras

5.3. Nuevos elementos

<meter>

Define los siguientes atributos:

- ✓ value: representa el valor actual. Si no se especifica, se toma como valor el primer número que aparece en el contenido del elemento. Su valor por defecto es 0
- ✓ min: el mínimo valor permitido. El valor por defecto es 0.
- ✓ max: el mayor valor permitido. El valor por defecto es 1, si min es mayor que 1, max será igual a min
- ✓ low: es considerado el límite inferior del rango de valores
- ✓ max: es considerado el límite superior del rango de valores
- ✓ optimum: representa el valor óptimo del elemento, y se encuentra entre min y max

5.4. Ejercicio práctico

Crear un formulario que contenga lo siguiente:

- ✓ Los 12 nuevos tipos de elementos input.
- ✓ El nuevo elemento datalist, que contenga algunos nombres de provincia y un campo de texto que se relacione con él.
- ✓ Una caja de texto (`<input type="text">`), a la cual aplicar los atributos autofocus, placeholder, required y autocomplete.
- ✓ Una caja de texto que sólo pueda contener números (`<input type="number">`), cuyos valores tienen que estar comprendidos entre 0 y 10.
- ✓ Un campo de selección de ficheros (`<input type="file">`), al que aplicar el atributo multiple.
- ✓ Un campo de introducción de password (`<input type="password">`), donde el valor introducido debe cumplir lo siguiente: debe tener una longitud mínima de 8 caracteres, comenzar por un número y terminar en una letra mayúscula.
- ✓ Un nuevo elemento progress que represente el avance de completado de campos del formulario.

Acceder al formulario desde al menos 4 navegadores (2 de escritorio y 2 de dispositivos móviles), y comprobad el comportamiento y funcionamiento en cada elemento del formulario. Anotad dichos resultados en una hoja de cálculo para futuras referencias.

6. Modernizr

Es una librería JavaScript que nos permite conocer la compatibilidad del navegador con tecnologías HTML5 y CSS3

Esto nos permitirá desarrollar sitios web que se adaptados a las capacidades de los distintos navegadores, o los más utilizados

Sabiendo que nuestro navegador soporta ciertas capacidades de CSS3 o de HTML5, podremos utilizarlas con libertad

Si sabemos que un navegador no es compatible con determinada funcionalidad, podremos implementar variantes o “hacks”

6.1. Cómo utilizar Modernizr

Descargar el archivo con el código fuente de Modernizr

<http://modernizr.com/download/>

Se trata de un archivo con código JavaScript que podemos encontrar en dos variantes:

- ✓ Development: contiene el código fuente completo, sin comprimir y con comentarios.
Se utiliza en desarrollo o si queremos acceder a su código para comprenderlo o ampliarlo
- ✓ Production: es OBLIGATORIO utilizar esta variante cuando pasamos a un entornos de producción

6.1. Cómo utilizar Modernizr

Al descargar Modernizr, tenemos la posibilidad de generar una librería únicamente con las funcionalidades que queremos detectar, esto nos permite ahorrar la carga de KB innecesarios

Se incluirá como cualquier otro fichero JavaScript

```
<script src="/js/lib/vendor/modernizr-custom.min.js"></script>
```

Se aconseja colocar el script dentro del HEAD, porque debe cargarse antes del BODY de la página, debido a un componente que quizás utilicemos, para permitir HTML5 en Internet Explorer, llamado HTML5 Shiv

6.1. Cómo utilizar Modernizr

Además, se recomienda colocarlo después de los estilos CSS para evitar un comportamiento poco deseable llamado FOUC, por el cual puede mostrarse, por un pequeño espacio de tiempo, la página sin los estilos CSS aplicados

Una vez incluida la librería, tendremos disponibles los scripts de detección de funcionalidades además de una serie de clases CSS que nos ayudarán a aplicar estilos solo cuando los navegadores los soporten

6.2. El objeto Modernizr

Una vez tenemos Modernizr cargado, se crea automáticamente un objeto JavaScript que tiene una serie de propiedades (booleanas) que nos indican si están o no disponibles cada una de las funcionalidades presentes en CSS3 y HTML5

```
if (Modernizr.boxshadow) {  
    // Podemos aplicar sombras de CSS3  
} else {  
    // La propiedad box-shadow no está disponible  
}
```

```
if (Modernizr.canvas) {  
    // Podemos utilizar canvas de HTML5  
} else {  
    // El elemento canvas no está disponible  
}
```

6.2. El objeto Modernizr

El listado completo de propiedades del objeto para la detección de funcionalidades HTML5 y CSS3 se puede encontrar en la propia documentación de Modernizr:

<http://modernizr.com/docs/>

6.3. Clases CSS de Modernizr

Cuando tenemos Modernizr cargado en nuestra página, éste crea automáticamente una serie de clases que asigna al elemento html del documento

Cada una de estas clases hace referencia a las características que soporta el navegador, permitiendo desde CSS adaptar la interfaz según las funcionalidades:

```
<html lang="en" class=" js flexbox canvas canvastext webgl no-touch geolocation postmessage  
websqldatabase indexeddb hashchange history draganddrop websockets rgba hsla multiplebgs  
backgroundsize borderimage borderradius boxshadow textshadow opacity cssanimations  
csscolumns cssgradients cssreflections csstransforms csstransforms3d csstransitions fontface  
generatedcontent video audio localstorage sessionstorage webworkers applicationcache svg  
inlinesvg smil svgclippaths">
```

....

```
</html>
```


6.3. Clases CSS de Modernizr

La creación de cada una de esas clases se realizará únicamente en caso de que el navegador sea compatible, de forma que luego podamos utilizar las clases de manera sencilla.

Por ejemplo, para aplicar sombra a un elemento con CSS3 en los navegadores que lo permitan y emular ese sombreado en los navegadores que no soporten el atributo box-shadow:

```
.elemento{  
  border-left: 1px solid #ccc;  
  border-top: 1px solid #ccc;  
  border-bottom: 1px solid #666;  
  border-right: 1px solid #666;  
}
```

6.3. Clases CSS de Modernizr

Si nuestro navegador es compatible con el atributo box-shadow de CSS3, Modernizr habrá incluido la clase "boxshadow" en el elemento html

Nosotros podemos utilizar dicha clase CSS para aplicar estilos que sabemos que solo acabarán afectando a los navegadores que soporten el atributo box-shadow

```
.boxshadow .elemento{  
  border: 1px solid #ccc;  
  box-shadow: #999 3px 3px 3px;  
}
```

6.4. Ejercicio práctico

Identificar las siguientes características de al menos 4 navegadores (2 de escritorio y 2 de dispositivos móviles):

- ✓ Cuáles de los 12 nuevos tipos de input soporta el navegador
- ✓ Qué codecs de reproducción de vídeo soporta cada navegador
- ✓ Qué sistema(s) de almacenamiento local soporta cada navegador

6.5. Modernizr.load()

El método `Modernizr.load()` es una sencilla manera para cargar librerías sólo cuando los usuarios las necesitan, normalmente, cuando una funcionalidad en concreto está (o no) disponible

Es una buena manera de ahorrar ancho de banda y mejorar un poco más el rendimiento de la aplicación

Por ejemplo, podemos saber si el navegador ofrece soporte al API de geolocalización (`Modernizr.geolocation`)

Podemos cargar unos recursos u otros dependiendo de la disponibilidad de esta funcionalidad con el método `Modernizr.load()`, que permite indicar a los navegadores que no soporten ese API que carguen el polyfill correspondiente

6.5. Modernizr.load()

Un “polyfill” es un trozo de código o un plugin que permite tener las nuevas funcionalidades de HTML5 en aquellos navegadores que nativamente no lo soportan

La sintaxis de Modernizr.load() es bastante sencilla de comprender:

```
Modernizr.load({  
  test: Modernizr.geolocation,  
  yep : 'geo.js',  
  nope: 'geo-polyfill.js'  
});
```

6.6. Ejercicio práctico

Identificar si el navegador soporta el atributo placeholder. En caso de no soportar dicha funcionalidad, cargar el polyfill correspondiente para añadir dicha funcionalidad al navegador