Desarrollo de aplicaciones multiplataforma

Elaborado por: José Antonio Sánchez

Módulo 3

Programación HTML5

Innova4B

SESIÓN 3

Innova4B

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.
- Utilizar un elemento meter, para simular la ocupación de un disco duro de 1TB, suponiendo que un grado alto de ocupación sea 800GB

5.4. Ejercicio práctico

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

Anotar dichos resultados en una hoja de cálculo para futuras referencias

Probar las funcionalidades con un IE7 e IE8:

http://utilu.com/IECollection/

Innova4B

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
}
```

Innova4B

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>

Innova4B

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

Innova4B

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 geolocalizacion (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'
});
```

Innova4B

6.6. Ejercicio práctico

Identificar si el navegador soporta el atributo placeholder y la etiqueta progress. En caso de no soportar dichas funcionalidades, cargar los “polyfill” correspondiente para añadir dicha funcionalidad al navegador

Probar que está funcionando el ejercicio con navegadores que no soportan placeholder y progress

Polyfills:
https://github.com/Modernizr/Modernizr/wiki/HTML5-Cross-browser-Polyfills

Innova4B

6.7. Ejemplo con formularios

https://github.com/zoltan-dulac/html5Forms.js

Permite la utilización de inputs de tiempo y color

Utiliza Modernizr para cargar las librerías que le hagan falta en cada caso