Grado en Ingeniería Informática del Software

Software y estándares para la Web

P8. APIS DE HTML5 Y SERVICIOS WEB DE CARTOGRAFÍA

Escuela de Ingeniería Informática
Universidad de Oviedo

Contenido

Objetivos

En esta práctica se va a realizar:

- La creación de un documento de script utilizando el estándar ECMAScript
- La creación de objetos para la realización de diferentes tareas en combinación con HTML.
- La generación de código HTML desde el documento de script utilizando la biblioteca jQuery y su posterior inserción en el documento HTML original.
- El uso de APIs de HTML5 para la realización de diferentes tareas que complementen la funcionalidad de un ejercicio.
- El consumo de servicios web de cartografía en combinación con los ficheros creados durante las prácticas de XML para la representación de la información sobre un mapa.
- La validación del código HTML estático y el código HTML generado

IMPORTANTE: Aquellos ficheros en los que se incluyan mapas de los diferentes proveedores de cartografía pueden tener errores de marcado HTML y de accesibilidad por la propia estructura de los mapas. Estos errores están permitidos, pero deben justificarse en la entrega de los ejercicios.

IMPORTANTE: Recuerda las pautas de trabajo establecidas en la primera sesión de prácticas (P0. Pautas de trabajo): valida todos los documentos HTML, valida todas las hojas de estilo CSS, comprueba la adaptabilidad y la accesibilidad con las herramientas proporcionadas.

Temática del proyecto: Escritorio virtual (Virtual Desktop)

El proyecto Escritorio Virtual (Virtual Desktop) es evolutivo y será creado, completado y modificado en las diferentes prácticas de la asignatura.

Ejercicio 1: Introducción al API File

Las APIs de HTML5 son elementos incluidos en el estándar HTML5 que dan acceso a una serie de funcionalidades comunes que ofrecen servicios de uso común al desarrollador. Entre estos servicios se encuentran los siguientes: acceso a ficheros para leer su contenido, uso de la pantalla completa y almacenamiento de variables en el navegador para uso durante la sesión del usuario.

En este ejercicio se utilizará API File de HTML5 para leer un archivo desde la máquina cliente que contiene la información de un conjunto de noticias y, posteriormente, mostrar esa información en un documento HTML.

El fichero de partida *noticias.txt* se encuentra disponible en el campus virtual dentro de la carpeta donde se incluyen los guiones de prácticas del bloque de ECMAScript. Este fichero contiene, en cada línea, la información de una noticia: titular, entradilla, texto de la noticia y autor; se utiliza el carácter "_" (sin comillas) como separador de cada dato de información del elemento.

NOTA: Se permite el uso de la biblioteca jQuery para la realización de este ejercicio.

Tarea 1. Creación de un nuevo documento JavaScript

Dentro de la carpeta js del directorio del proyecto Escritorio Virtual crea un nuevo fichero llamado noticias.js.

Añade una referencia a este nuevo fichero en el documento noticias.html.

En el caso de que se vaya a utilizar la biblioteca jQuery para resolver el ejercicio, añade al documento html mencionado anteriormente la referencia que permita importar la biblioteca jQuery.

Tarea 2. Creación de la clase Noticias

Dentro del fichero creado en la tarea anterior crea la clase Noticias.

Añade a esta clase un método constructor que compruebe si el navegador que está usando el usuario soporta el uso de la API File.

Crea en esta clase un método llamado *readInputFile* que sea el encargado de realizar la lectura del fichero *noticias.txt*

Guía para resolver la tarea 2

Consulta el ejemplo incluido a continuación para entender de qué forma se puede comprobar el soporte que ofrece el navegador a determinadas características del objeto Window.

http://di002.edv.uniovi.es/~cueva/JavaScript/62API-FILE-LeerArchivoTexto.html

Tarea 3. Lectura del contenido del fichero

Modifica el fichero noticias.html para incluir el input que permita recibir un fichero desde la máquina cliente. Configura dicho marcado para que se realice una invocación al método *readInputFile* de la clase Noticias cuando el cliente cargue el archivo en el input.

Modifica el método *readInputFile* de la clase Noticias para que lea el fichero obtenido desde el input línea a línea y cree en el html la estructura que permita consultar los elementos incluidos en el archivo.

Guía para resolver la tarea 3

Consulta el ejemplo incluido a continuación para comprobar cómo se debe configurar un input de html para que reciba un único fichero desde la máquina cliente e invoque a un método cuando haya un cambio en el input.

Además, consulta el código del método del código leerArchivoTexto, que es el encargado de leer el contenido del fichero.

http://di002.edv.uniovi.es/~cueva/JavaScript/62API-FILE-LeerArchivoTexto.html

Utiliza bien los métodos de la librería jQuery o bien los métodos de la clase document para crear en el html el marcado que permita representar noticia incluida en el fichero que se ha leído.

Considera utilizar métodos predefinidos de ECMAScript como apoyo en la realización de las diferentes partes de la tarea; por ejemplo, el troceado de cada línea del fichero a partir del carácter separador mencionado en el enunciado.

Tarea 4. Inserción de nueva información

Añade al fichero noticias.html los elementos de html necesarios para que el usuario pueda añadir sus propias noticias al conjunto de noticias leídas desde el fichero. Cada noticia que se añada debe ubicarse al final del conjunto de noticias ya existente.

Modifica la clase Noticias para crear un método que añada la información de la nueva noticia al conjunto de noticias ya existente.

Guía para resolver la tarea 4

Modifica el documento html para añadir varios inputs que permitan recoger los diferentes elementos que componen la noticia. Incluye en esta modificación un botón que inserte en el html el contenido de la noticia creada a mano por el usuario cuando reciba una pulsación.

Utiliza bien los métodos de la librería jQuery o bien los métodos de la clase document para crear el bloque con la información relativa a la nueva noticia.

Resultado del ejercicio 1

Una vez completado el ejercicio deberían haberse añadido los siguientes ficheros al proyecto del Escritorio Virtual:

- Fichero noticias.js en el directorio js del proyecto

- Fichero noticias.css en el directorio estilo del proyecto

Además, se debe haber modificado el fichero noticias.html para incluir las referencias necesarias para el funcionamiento del código y el input que permita leer el fichero desde la máquina cliente.

Ejercicio 2: Introducción a la API de Geolocalización y los mapas

En este ejercicio se utilizará la API de Geolocalización de HTML5 para acceder a la información de la posición del usuario.

Es posible utilizar las siguientes soluciones como proveedor de mapas: Google Maps (https://www.google.es/maps/preview) y MapBox (https://www.mapbox.com). En caso de querer utilizar otra alternativa, se debe consultar con los profesores de la asignatura.

NOTA: Para que la obtención de la información de geolocalización funcione al 100% debe usarse el protocolo HTTPS y permitir al navegador el acceso a la información de la geolocalización.

NOTA: Se permite el uso de la biblioteca jQuery para la realización de este ejercicio.

Tarea 1. Creación del documento viajes.js

Dentro de la carpeta js del directorio del proyecto Escritorio Virtual crea un nuevo fichero llamado viajes.js.

Añade una referencia a este nuevo fichero en el documento viajes.html.

Tarea 2. Creación de la clase Viajes

Dentro del fichero creado en la tarea anterior crea la clase Viajes.

Añade a esta clase un método constructor que obtenga la posición geográfica del usuario y la almacene en diferentes atributos dentro de la clase.

Guía para resolver la tarea 2

Consulta el siguiente enlace para conocer la manera en la que es posible obtener los datos de geolocalización del usuario.

https://di002.edv.uniovi.es/~cueva/JavaScript/104-ClaseGeolocalizacion.html

Tarea 3. Manejo de errores de geolocalización en la clase Viajes

Modifica la clase Viajes para añadir el manejo de errores relacionados con la geolocalización del usuario.

Guía para resolver la tarea 3

Compara el código del enlace de la guía de la solución de la tarea anterior con el del siguiente ejemplo para comprender como se añade gestión de errores a la obtención de la información de geolocalización.

https://di002.edv.uniovi.es/~cueva/JavaScript/105-ClaseGeolocalizacionManejoErrores.html

Tarea 4. Creación de un mapa estático

Utiliza el proveedor de mapas escogido para obtener un mapa estático que represente la posición del usuario, obtenida usando la geolocalización, en el centro de dicho mapa.

Incluye el mapa estático obtenido en el fichero viajes.html

Guía para resolver la tarea 4

Consulta el siguiente enlace para entender cómo se utiliza un proveedor de mapas para obtener un mapa estático cuyo punto central esté en un lugar concreto.

https://di002.edv.uniovi.es/~cueva/JavaScript/106-ClaseMapaEstaticoGoogle.html

Tarea 5. Creación de un mapa dinámico

Utiliza el proveedor de mapas escogido para obtener un mapa dinámico que represente la posición del usuario obtenida usando la geolocalización.

Incluye el mapa estático obtenido en el fichero viajes.html

Guía para resolver la tarea 5

Consulta el siguiente enlace para entender cómo se utiliza un proveedor de mapas para obtener un mapa dinámico cuyo punto central esté en un lugar concreto.

https://di002.edv.uniovi.es/~cueva/JavaScript/108-GeolocalizacionMapaDinamicoGoogle.html

Tarea 6. Procesado de un fichero en formato XML con el API File

Utilizando jQuery y el API File de html, crea un control en el fichero viajes.html que permita subir **un único archivo** desde la máquina cliente en formato XML y muestra la información completa de dicho fichero en formato HTML.

NOTA: Utiliza el fichero rutasEsquema.xml creado durante las prácticas del módulo de XML de la asignatura como fichero de entrada para esta tarea.

Guía para resolver la tarea 6

Consulta el siguiente enlace para entender cómo se utiliza un control de html para permitir la subida de un archivo desde la máquina cliente para que después pueda ser procesado.

http://di002.edv.uniovi.es/~cueva/JavaScript/62API-FILE-LeerArchivoTexto.html

Tarea 7. Lectura y procesamiento de archivos de planimetría

Utilizando jQuery y el API File de html, crea un control en el fichero viajes.html que permita subir **varios archivos** desde la máquina cliente en formato KML y representa la información de dichos ficheros en un mapa dinámico.

NOTA: Utiliza todos los ficheros de planimetría en formato KML creados durante las prácticas del módulo de XML de la asignatura como ficheros de entrada.

Guía para resolver la tarea 7

Consulta el siguiente enlace para entender cómo se utiliza un control de html para permitir la subida de varios archivos a la vez desde la máquina cliente para que después puedan ser procesados.

http://di002.edv.uniovi.es/~cueva/JavaScript/63API-FILE-ArchivosMultiples.html

Consulta la documentación del proveedor de mapas escogido para conocer cómo se puede utilizar la información de este tipo de ficheros para mostrar puntos superpuestos en un mapa.

https://developers.google.com/maps/documentation/javascript/kml?hl=es-419

https://docs.mapbox.com/mapbox-gl-js/example/

Tarea 8. Lectura y procesamiento de archivos de altimetría

Utilizando jQuery y el API File de html, crea un control en el fichero viajes.html que permita subir **varios archivos** desde la máquina cliente en formato SVG y representa la información de dichos ficheros en el documento HTML.

NOTA: Utiliza todos los ficheros de altimetría (perfiles) en formato SVG creados durante las prácticas del módulo de XML de la asignatura como ficheros de entrada.

Resultado del ejercicio 2

Una vez completado el ejercicio deberían haberse añadido los siguientes ficheros al proyecto del Escritorio Virtual:

- Fichero viajes.js en el directorio js del proyecto
- Fichero viajes.css en el directorio estilo del proyecto

Además, se debe haber modificado el fichero viajes.html para incluir las referencias necesarias para el funcionamiento del código.

Ejercicio 3: Ejercicio sobre APIs de HTML5

En este ejercicio se va a escribir una aplicación web, utilizando HTML5, CSS y JavaScript. La aplicación web es de **temática libre**, pero **debe usar como mínimo 3 API de HTML5.**

Se valorará la presentación, la complejidad de la aplicación, la originalidad, la creatividad y las API de HTM5 utilizadas.

NOTA: Las APIs Fullscreen y Canvas no podrán utilizarse juntas para la resolución de este ejercicio.

NOTA: Se permite el uso de la biblioteca jQuery para la realización de este ejercicio.

Guía para resolver el ejercicio 3

Algunos ejemplos de API de HTML5

- http://di002.edv.uniovi.es/~cueva/JavaScript/22Canvas.html
- https://di002.edv.uniovi.es/~cueva/JavaScript/24Geolocalizacion.html
- http://di002.edv.uniovi.es/~cueva/JavaScript/62API-FILE-LeerArchivoTexto.html

Listado de APIS de HTML5

- **API TextTrack.** Acceso a las pistas de elementos multimedia (video o audio)
- **API Fullscreen.** Manejo de pantalla completa.
- **API Stream.** Acceso a contenido multimedia de flujo continuo o en streaming.
- **API Canvas.** Permite dibujar, animar y procesar imágenes.
- **API WebGL.** Permite crear gráficos 3D para la Web. Suele acompañarse de la biblioteca Three.js (www.threejs.org)
- **API Pointer Lock.** Facilita la interacción con el puntero del ratón.
- **API Drag and Drop.** Facilita arrastrar y soltar en la Web
- **API Web Storage.** Es básicamente una mejora de las cookies. Permite almacenar datos en el disco duro del usuario y utilizarlos posteriormente.
- **API IndexedDB.** Permite crear una pequeña base de datos indexada en el ordenador del usuario.
- **API File.** Permite el manejo de archivos.
- **API Geolocation.** Permiten determinar la ubicación física real del usuario.
- **API History.** Permite gestionar el registro histórico de navegación.
- **API Offline.** Facilita la navegación cuando está el usuario fuera de línea.
- **API Page Visibility.** Informa a la aplicación sobre el estado de visibilidad del documento. Por ejemplo cuando se minimiza una ventana.
- **API Web Messaging.** Permite que aplicaciones de orígenes diferentes se comuniquen entre sí.
- **API WebSocket.** Permite la conexión del cliente con el servidor de una forma más rápida y eficaz a través de TCP sin enviar cabeceras HTTP.
- **API WebRTC.** Permite la comunicación en tiempo real.
- **API Web Audio.** Permite el procesamiento de audio.
- **API Web Workers.** Permite ejecutar tareas en segundo plano.

- **API Clipboard.** Permite el manejo del portapapeles.
- **API Device Orientation.** Permiten determinar la orientación y el movimiento del dispositivo.
- **API Quota Management.** Permite comprobar el espacio de almacenamiento disponible.
- **API SVG.** Permite generar gráficos vectoriales según el estándar SVG

Resultado del ejercicio 3

Una vez completado el ejercicio deberían haberse añadido los siguientes ficheros al proyecto del Escritorio Virtual:

- Fichero api.js en el directorio js del proyecto
- Fichero api.css en el directorio estilo del proyecto
- Fichero api.html en el directorio raíz del proyecto

Además, aunque no sea un juego propiamente dicho, se debe haber modificado el fichero juegos.html para añadir un enlace al menú de juegos que permita acceder al resultado de este ejercicio.

Recuerda

Se requiere el uso correcto de los elementos HTML5 establecidos en los ejercicios y se valorará positivamente el uso correcto y adecuado al contexto y funcionalidad, de elementos adicionales HTML5.

Se requiere el uso correcto de las propiedades de los módulos CSS establecidos en los ejercicios y se valorará positivamente el uso correcto y adecuado al contexto, de propiedades y módulos adicionales CSS.

Solamente se permite el paradigma de orientación de objetos y todo debe estar organizado con clases y objetos. Además, solamente se permite el uso de la biblioteca jQuery para realizar las llamadas a los servicios externos y la creación de contenido en el html, quedando prohibido el uso de cualquier otra biblioteca.

IMPORTANTE: Todas las llamadas que se realicen a la biblioteca jQuery deben estar encapsuladas en los métodos pertenecientes a las clases que se utilicen para resolver los ejercicios.

Las siguientes condiciones son de obligado cumplimiento en todo el proyecto:

- **TODOS** los documentos HTML que componen el proyecto deben ser HTML5 válidos y sin advertencias utilizando el validador de lenguajes de marcado del W3C.
    - Recuerda que el contenido de los documentos HTML puede cambiar después de la ejecución del código ECMAScript. En ese caso, se debe comprobar la validez del código HTML en todos los diferentes estados por los que pase el documento.
- No está permitido el uso indiscriminado de bloques anónimos (**div**) en los documentos HTML5, se debe priorizar el uso de contenedores semánticos HTML5.
- Se deben utilizar selectores específicos, el uso de selectores id y class deberá estar justificado a través de comentarios en las hojas de estilo que expliquen la necesidad de su utilización y para que se utilizan.
- **TODAS** las reglas de todas las hojas de estilo deben estar precedidas por un comentario donde se indique la especificidad del (o los) selectores de la regla.
- **TODAS** las hojas de estilo que se utilizan en el sitio web deben ser validas utilizando el validador CSS del W3C
- **TODAS** las hojas de estilo deben tener 0 advertencias.
    - Recuerda seleccionar en “Más opciones” el informe de “Todas las advertencias” dentro de las opciones del validador CSS del W3C.
    - Excepcionalmente se permite las advertencias referidas a la verificación de los colores (color y background-color). **OBLIGATORIAMENTE** se debe indicar mediante un comentario en la regla de la hoja de estilo afectada la herencia de colores garantizando que la advertencia ha sido comprobada, verificada y garantizando que no provoca efectos laterales no deseados.
    - Excepcionalmente se permiten las advertencias referidas a la redefinición de propiedades derivadas del uso de @media-queries. **OBLIGATORIAMENTE** se debe indicar mediante un comentario en las reglas de la hoja de estilo afectada que propiedades se están redefiniendo.
- Se debe garantizar la adaptabilidad y realizar su verificación para todos los documentos que componen el proyecto.

    - Se deben utilizar medidas relativas en las hojas de estilo.
- Se debe garantizar la accesibilidad del proyecto mediante los test de las herramientas de accesibilidad para el nivel AAA de las WCAG 2.0 con 0 errores de modo automático en todos los documentos que lo componen.

El no cumplimiento de las características anteriores derivará en la invalidación del proyecto.