

Software y estándares para la Web

P4. COMPUTACIÓN EN EL CLIENTE (SESIÓN 3: API HTML5)

Contenido

Objetivos y requisitos comunes a todos los ejercicios.....	2
Ejercicio 12	2
Ejercicio 13	3
Ejercicio 14	3
Ejercicio 15	4

RECUERDA:

La entrega es en el mismo archivo empaquetado que la sesión 1 y la sesión 2.

El archivo empaquetado tiene que tener las carpetas Ejercicio-12, Ejercicio-13,etc. En su interior estarán los archivos solicitados

Objetivos y requisitos comunes a todos los ejercicios

En esta práctica el objetivo es hacer computación en el cliente Web usando: las bibliotecas del **API HTML5**.

Se usará solamente el **paradigma de orientación a objetos**.

Los archivos HTML, CSS y JavaScript deben estar siempre separados y no incrustados en el archivo HTML. El objetivo es tener en archivos separados el contenido, la presentación y la computación.

Se debe comprobar la validez de los archivos contruidos HTML5 y CSS con los validadores del W3C.

Los ejercicios desarrollados deben funcionar en todos los navegadores de referencia: Firefox, Edge, Chrome, Opera, Safari en los Mac e IOS, Chrome en Android, etc.

Esta práctica se corresponde con los temas de teoría:

- Computación Web
- Lenguajes de Script
- El lenguaje JavaScript
- Tecnologías y recursos relacionados con JavaScript

Ejercicio 12

Tarea 1. Utilizando el API de HTML5 denominado **Geolocation**, escribir una clase denominada **GeoLocalizacion** que nos informe de la posición del usuario.

Para utilizar la geolocalización del usuario:

- Debe usarse el protocolo https.
- Los servidores desde donde se ejecuta la geolocalización pueden dar avisos en los distintos navegadores de que sus certificados no están validados. Debe hacerse una excepción y permitir cargar el sitio Web
- El usuario debe aceptar que sea geo-localizado, debido a que su situación es un dato de carácter personal.

Tarea 2. Se debe ampliar el código JavaScript de la clase **GeoLocalizacion** para que permita:

- Manejo de errores de geo-localización.

Tarea 3. Se debe ampliar el código JavaScript de la clase **GeoLocalizacion** para que permita:

- Mostrar un mapa estático en Google Maps con un marcador con la posición del usuario.

Tarea 4. Se debe crear un objeto en JavaScript usando el **API de Google Maps** para que permita:

- Crear un mapa dinámico con un determinado centro. El centro debe señalarse con un marcador

Tarea 5. Se debe crear un objeto en JavaScript usando el **API de Google Maps** para que permita:

- Crear un mapa dinámico con Google Maps indicando la posición del usuario

Tarea 6. Se debe crear una aplicación de temática libre en JavaScript usando el **API de Google Maps**.

- Puede combinarse con el uso de otras APIs.

Epílogo. Todas las tareas se presentarán en distintas sub-carpetas denominadas **Tarea-1, Tarea-2, etc.** Dentro de la carpeta denominada **Ejercicio-12**.

Ejercicio 13

Tarea 1.

Escribir un archivo en HTML5 denominado **Ejercicio13** (con extensión **.html**) que referencie a un archivo CSS denominado **Ejercicio13** (con extensión **.css**) y también debe referenciar a un archivo en JavaScript (extensión **.js**).

Se debe escribir una aplicación que use el **API FILE de HTML5** que **cargue archivos desde la máquina cliente y que visualice sus propiedades (tamaño, tipo, etc.)**.

En el caso de que se seleccionen archivos de tipo:

- **Texto**
- **JSON**
- **XML**

Deben mostrarse íntegramente por pantalla.

El diseño de la presentación se deja libre al estudiante.

Epílogo. Todas las tareas se presentan en la misma sub-carpeta denominada **Ejercicio-13** que debe contener los archivos:

- Ejercicio13.html
- Ejercicio13.css
- Ejercicio13.js
- Ejemplo.txt
- Ejemplo.json
- Ejemplo.xml

Ejercicio 14

Tarea 1. Escribir un archivo en HTML5 denominado **MapaKML** (con extensión **.html**) que referencie a un archivo CSS denominado **MapaKML** (con extensión **.css**) y también debe referenciar a uno o varios archivos en JavaScript (extensión **.js**).

Se debe escribir una aplicación usando el **API de Google Maps** que **cargue desde la máquina cliente el archivo KML de rutas turísticas creado en la práctica de XML**.

Debe incluirse el archivo KML de rutas turísticas utilizado.

El diseño de la presentación se deja libre al estudiante.

Tarea 2. Escribir un archivo en HTML5 denominado **MapaGeoJSON** (con extensión **.html**) que referencie a un archivo CSS denominado **MapaGeoJSON** (con extensión **.css**) y también debe referenciar a uno o varios archivos en JavaScript (extensión **.js**).

Se debe escribir una aplicación usando el **API de Google Maps** que **cargue desde la máquina cliente un archivo en formato GeoJSON** con la situación del inicio y de los hitos de las rutas turísticas construidas en la práctica de XML.

Debe buscarse el estándar GeoJSON y crear un archivo denominado **rutas.GeoJSON**. Este archivo es el que se cargará. Debe de incluirse obligatoriamente.

El diseño de la presentación se deja libre al estudiante.

Epílogo. Cada tarea se presenta en la sub-carpeta correspondiente. Debe haber dos sub-carpetas denominadas Tarea-1 y Tarea-2 dentro de la carpeta **Ejercicio-14**.

Ejercicio 15

Tarea 1. Realización de 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 HTML5 utilizadas.

Epílogo. Todo se presenta en la misma sub-carpeta denominada **Ejercicio-15**