

Práctica 4

11/11/2021

David Moreno Hidalgo
César Munuera Pérez
Víctor Fernández Torres
Victor Lapeña López
Pablo González Jalvo

• 2. Ejercicio JavaScript

En este primer ejercicio, creamos una página apta para camareros para tener un seguimiento más sencillo de las comandas. Podrán seleccionar entre los primeros, segundos y postres, además de los añadidos del café y demás. Además, se mantendrán los pedidos en caso de cambiar de mesa. Encontraremos la opción de pagar, por la cual se realizará un cálculo de los platos elegidos y los suplementos.

La clase main se han estructurado de la siguiente forma:

- En la primera parte encontramos la declaración de variables, estas serán: platosPrincipales, preciosPlatosPrincipales, platosSegundos, preciosPlatosSegundos, platosPostres, preciosPlatosPostres; así como los Platos pedidos, las mesas y la opción de café o copa.
- La siguiente parte son las funciones, en los que encontramos: InicializarArrayPlatos: primero vacía el array de platos principales y luego recorre rellenando con los platos principales.

CalcularPrecio: El cual calcula los precios de los platos cuando se seleccionan, este está inicializado a 0 y mediante contadores, actualiza el precio.

funcionMesa: Que selecciona las distintas posibles mesas

- Después, el campo correspondiente a las jQuerys: #radio1, #radio2 y #radio3 componen la lógica para cambiar entre las mesas.

#slct1 es usado para que, al cambiar de mesa, se mantenga la selección de platos y que, al volver, se siga mostrando los platos.

#CajaPlatosPedidos es donde se almacenan las selecciones.

#CajaMenu permite almacenar los platos cuando se eligen.

- Por último, el código del programa
Este inicializa todo el programa, en la clase “restaurante” tenemos un enlace a las librerías de JQuery.

Aquí ya encontramos cómo se han implementado los radios

Por último, encontramos el botón pagar, el cual calcula los precios de los platos seleccionados y los muestra por pantalla.

- ☐ Primeros platos
☐ Segundos platos
☒ Postres

Mesa 1 ▼

Tarta Yogur Helado Manzana	Tarta Pollo Macarrones
-------------------------------------	------------------------------

- ☒ Con café
☐ Con copa

Pagar

- 3. Ejercicios con el Lenguaje HTML5
 - 3.1 Comprobación de funcionalidades en navegadores

Características	Opera 80.0	Chrome 95.0.4638.69	Dev 95	Edge	Firefox 94.0
HTML5 tokenizer	Si	Si		Si	Si
HTML5 tree building	Si	Si		Si	Si
Parsing inline SVG	Si	Si		Si	Si
Parsing inline MathML	Si	Si		Si	Si
Embedding custom non-visible data	Si	Si		Si	Si
Section elements	Si	Si		Si	Si

Grouping content elements	Si	Si	Si	Si
Text-level semantic elements	Si	Si	Si	Parcial
Interactive elements	Si	Si	Parcial	Parcial
hidden attribute	Si	Si	Si	Si
hidden attribute	Si	Si	Si	Si
input type=search	Si	Si	Si	Si
input type=tel	Si	Si	Si	Si
input type=url	Si	Si	Si	Si
input type=email	Si	Si	Si	Si
input type=date	Si	Si	Si	Si
input type=month	Si	Si	Si	No
input type=week	Si	Si	Si	No
input type=time	Si	Si	Si	Si
input type=datetime-local	Si	Si	Si	Si
input type=number	Si	Si	Si	Si
input type=range	Si	Si	Si	Si
input type=color	Si	Si	Si	Si
input type=checkbox	Si	Si	Si	Si
input type=image	Si	Si	Si	Si
input type=file	Parcial	Parcial	Parcial	Parcial
textarea	Si	Si	Si	Si
select	Si	Si	Si	Si
fieldset	Si	Si	Si	Si
datalist	Si	Si	Si	Si
output	Si	Si	Si	Si
progress	Si	Si	Si	Si
meter	Si	Si	Si	Si
Field validation	Si	Si	Si	Si
Association of controls and forms	Si	Si	Si	Si
CSS selectors	Si	Si	Si	Si
Events	Si	Si	Si	Si

Form validation	Si	Si	Si	Si
Custom elements	No	No	No	No
Shadow DOM	Si	Si	Si	Si
HTML templates	Si	Si	Si	Si
HTML imports	No	No	No	No
Geolocation	Si	Si	Si	Si
Device Orientation	No	No	No	Si
Device Motion	No	No	No	Si
Full screen support	Si	Si	Si	Si
Web Notifications	Si	Si	Si	Si
Gamepad control	Si	Si	Si	No
Pointer Events	Si	Si	Si	Si
Pointer Lock support	Si	Si	Si	Si
Server-Sent Events	Si	Si	Si	Si
Beacon	Si	Si	Si	Si
Fetch	Si	Si	Si	Si
Upload files	Si	Si	Si	Si
Response type support	Si	Si	Si	Si
Basic socket communication	Si	Si	Si	Si
ArrayBuffer and Blob support	Si	Si	Si	Si
Readable streams	Si	Si	Si	Si
Writable streams	No	No	No	No
WebRTC 1.0	Si	Si	Si	Si
ObjectRTC API for WebRTC	No	No	No	No
Data channel	Si	Si	Si	Si
Access the webcam	No	No	No	No
Screen Capture	No	No	No	No
Enumerate devices	No	No	No	No
Media Stream recorder	Si	Si	Si	Si
draggable attribute	Si	Si	Si	Si
dropzone attribute	No	No	No	No

Editing elements	Si	Si	Si	Si
Editing documents	Si	Si	Si	Si
CSS selectors	Si	Si	Si	Si
APIs	Si	Si	Si	Si
Clipboard API and events	Si	Si	Si	Si
spellcheck attribute	Si	Si	Si	Si
Web Workers	Si	Si	Si	Si
Shared Workers	Si	Si	Si	Si
window.requestIdleCallback	Si	Si	Si	Si
Web Cryptography API	No	No	No	No
Content Security Policy 1	Si	Si	Si	Si
Content Security Policy 2	Si	Si	Si	Si
Cross-Origin Resource Sharing	Si	Si	Si	Si
Subresource Integrity	Si	Si	Si	Si
Cross-document messaging	Si	Si	Si	Si
Web Authentication / FIDO 2	No	No	No	No
Credential Management	No	No	No	No
Sandboxed iframe	Si	Si	Si	Si
iframe with inline contents	Si	Si	Si	Si
Web Payments	No	No	No	No
video element	Si	Si	Si	Si
Subtitles	Si	Si	Si	Si
Audio track selection	No	No	No	No
Video track selection	No	No	No	No
Poster images	Si	Si	Si	Si
Codec detection	Si	Si	Si	Si
MPEG-4 ASP support	No	No	No	No
H.264 support	Si	Si	Si	Si
H.265 support	Si	No	No	No
Ogg Theora support	Si	Si	Si	Si
WebM with VP8 support	Si	Si	Si	Si

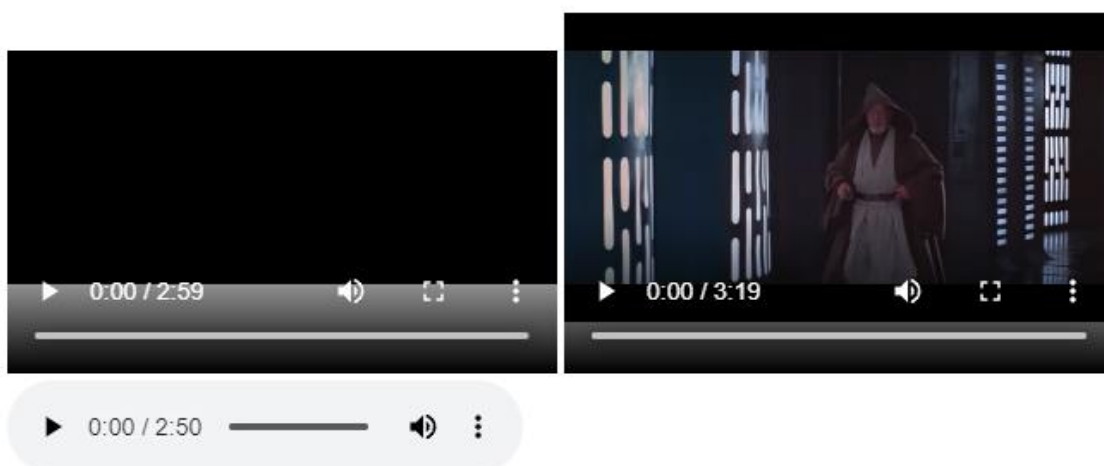
WebM with VP9 support	Si	Si	Si	Si
audio element	Si	Si	Si	Si
Loop audio	Si	Si	Si	Si
Preload in the background	Si	Si	Si	Si
Web Audio API	Si	Si	Si	Si
Speech Recognition	No	Si	Si	No
Speech Synthesis	Si	Si	Si	Si
PCM audio support	Si	Si	Si	Si
MP3 support	Si	Si	Si	Si
AAC support	Si	Si	Si	Si
Dolby Digital support	No	Si	Si	No
Dolby Digital Plus support	Si	Si	Si	No
Ogg Vorbis support	Si	Si	Si	Si
Ogg Opus support	Si	Si	Si	Si
WebM with Vorbis support	Si	Si	Si	Si
WebM with Opus support	Si	Si	Si	Si
Media Source extensions	Si	Si	Si	Si
DRM support	No	No	No	Si
Dynamic Adaptive Streaming / MPEG-DASH	No	No	No	No
HTTP Live Streaming / HLS	No	No	No	No
Video codecs	Parcial	Parcial	Parcial	Parcial
Audio codecs	Parcial	Parcial	Parcial	Parcial
picture element	Si	Si	Si	Si
srcset attribute	Si	Si	Si	Si
sizes attribute	Si	Si	Si	Si
Canvas 2D graphics	Si	Si	Si	Si
Text support	Si	Si	Si	Si
Path support	Si	Si	Si	Si
Ellipse support	Si	Si	Si	Si
Dashed line support	Si	Si	Si	Si

System focus ring support	Si	Si	Si	Si
Hit testing support	No	No	No	No
Blending modes	Si	Si	Si	Si
PNG support	Si	Si	Si	Si
JPEG support	Si	Si	Si	Si
JPEG-XR support	No	No	No	No
WebP support	Si	Si	Si	No
WebGL	Si	Si	Si	Si
WebGL 2	Si	Si	Si	Si
WebVR	No	No	No	No
Web Animations API	Si	Si	Si	Si
window.requestAnimationFrame	Si	Si	Si	Si
Application Cache	No	No	No	No
Service Workers	No	No	No	No
Push Messages	Si	Si	Si	Si
Custom scheme handlers	No	No	No	No
Custom content handlers	No	No	No	No
Session Storage	Si	Si	Si	Si
Local Storage	Si	Si	Si	Si
IndexedDB	Si	Si	Si	Si
Objectstore Blob support	Si	Si	Si	Si
Web SQL Database	Si	Si	Si	No
Basic support for reading files	Si	Si	Si	Si
Create a Blob from a file	Si	Si	Si	Si
Create a Data URL from a Blob	Si	Si	Si	Si
Create an ArrayBuffer from a Blob	Si	Si	Si	Si
Create a Blob URL from a Blob	Si	Si	Si	Si
FileSystem API	No	No	No	No
Asynchronous script execution	Si	Si	Si	Si
Deferred script execution	Si	Si	Si	Si
Script execution events	No	No	No	Si

ECMAScript 5	Si	Si	Si	Si
ECMAScript 6	Si	Si	Si	Si
ECMAScript 7	Si	Si	Si	Si

○ 3.2. Creación de página con elementos multimedia

En este siguiente ejercicio creamos una página en la cual se verá reflejado 2 videos y un audio. En este archivo, también podemos definir las dimensiones de las ventanas de video



○ 3.3 Creación de página con formulario para su autovalidación

En este ejercicio diseñamos y creamos una página web en la cual se debe rellenar los datos de aquel que entra en esta, indicando su email, su fecha de nacimiento, tarjeta de crédito... La clase "Formulario Validación Tarjeta Credito" es el esqueleto de programa index, tiene las funciones necesarias para hacer posible el funcionamiento de los campos para rellenar los datos. De todas estas funciones, las tres últimas son las mas usadas, ya que estas son la fecha de expiración, el propio número de tarjeta y el CCV. La clase index 3.3 es el programa principal, el que da forma a lo que aparece en pantalla, con todos los campos declarados y la estructura de estos.

Introduce tu email:

Fecha nacimiento:

Número de Tarjeta

Fecha de Expiración

CCV

- **3.4 Ejercicio práctico de creación de páginas con otras funcionalidades**

Aquí encontramos varias implementaciones de HTML5.

Geolocalización

Aquí se ha utilizado la función “geoloc”. Además, se puede comprobar si el dispositivo que esta siendo usado, es compatible con la geolocalización, en todo caso, se abrirá un aviso. “navigator.geolocation.getCurrentPosition(showPosition)” es usado para obtener la posición exacta, mientras que “position.coords.latitude” “position.coords.longitude” son usados para obtener la altitud y la latitud.

[Ver geolocalización.](#)

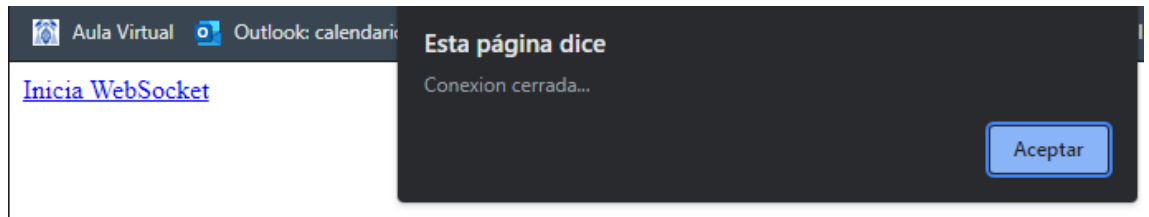
Tu dispositivo soporta la geolocalización.

Latitud: 40.4946944

Longitud: -3.3685504

Sockets

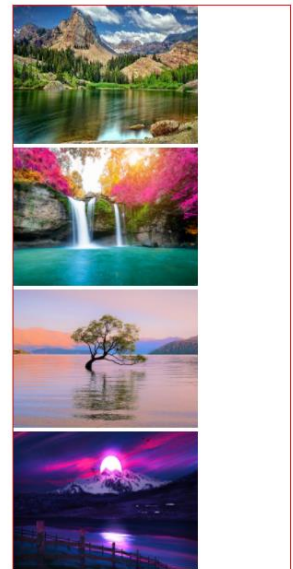
En esta página se envían unos mensajes al localhost, si se vuelve a pulsar “Inicia WebSocket” se cerrará la conexión. Esta página también cuenta con una alerta que dirá si websocket es compatible con el buscador usado. “ws.onopen” se usa para abrir el WebSocket, mientras que “ws.onclose” lo cierra. “ws.send” envía los mensajes.



Drop&Drag

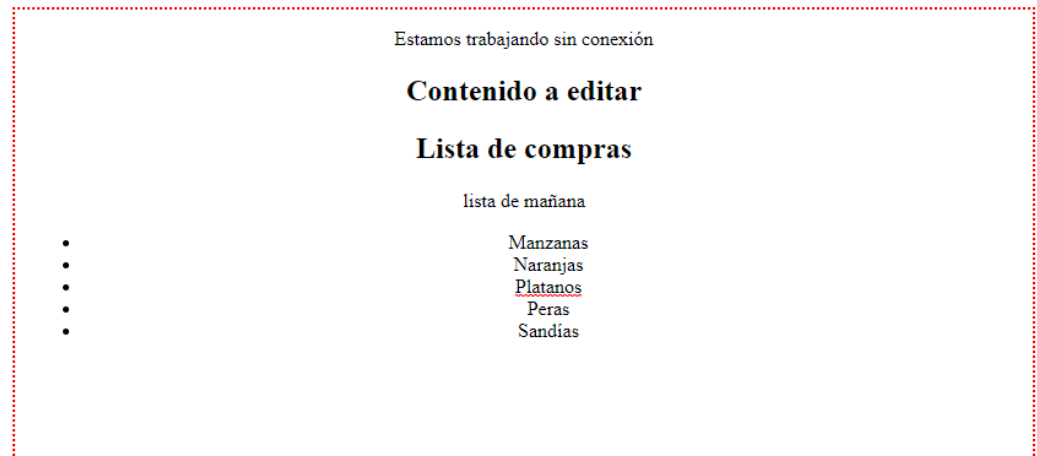
Con esta página web podemos arrastrar imágenes para poder visualizarlas más cómodamente. En el código se implementa un cuadro de imagen (donde se encuentran las imágenes a visualizar), la zona de destino de las imágenes. Se usan como eventos los “EventListener” para que puedan suceder las acciones y las demás funciones que hacen que la página trabaje con normalidad. Las funciones “comenzando_arrastras” permite poder mantener pulsada una imagen para desplazarla hasta el cuadro correspondiente. “Dragenter”, “dragover”, “drop” y “dragleave” son las funciones que componen toda la lógica de esta página web.

Página simple que aplica el funcionamiento drag & drop



Trabajo Offline

Esta última página nos permite trabajar sin necesidad de conexión a internet. Para ello, descarga en la caché. Para poder trabajar, se debe crear un archivo “manifiesto” que recoge todos los archivos desde el caché y guarda los cambios y los actualiza cuando tiene conexión.



○ 4. Extensión del Ejercicio JavaScript.

Ahora, a partir del ejercicio del punto 2, implementamos algunas de las funciones de los ejercicios del punto 3, en este caso, el formulario del pago con tarjeta y el poder trabajar online. La página principal “index” cuenta con el código del restaurante creado en el ejercicio 2, esta también tiene archivo JavaScript “main.js” y un archivo CSS y un archivo “Manifest” para poder trabajar offline. Tanto el main.js como el CSS y el Manifest se usan para “index” y para “popTarjetaCrédito”, mientras que el JavaScript “main”, únicamente para la “index”.

Por otro lado tenemos dos nuevas páginas, “popTarjetaCrédito” y “FormularioValidaciónTarjeta” en los que se encuentra el mismo código que en el apartado 3.3.

La página “index” contiene un nuevo botón (pagar) que es un enlace para ir acceder al pago con tarjeta. A su vez, “popTarjetaCrédito” también tiene un botón homónimo para regresar a la página “main”.

RESTAURANTE

- ☐ Primeros platos
- ☒ Segundos platos
- ☐ Postres

Pasa 1 v

Cocido
Puro
Macarrones
Ensalada



- ☐ Con café
- ☐ Con copa

1.2

Reservar pedido

Pagar

PAGAR CUENTA

Número de Tarjeta 1232 1231 1232 1231

Fecha de Expiración 05 2025

CCV 123

Pagar