

Práctica 7: JavaScript

Ejercicio 1

Página de compra con jQuery

En un sitio web de venta de bicicletas y accesorios se desea, como parte del proceso de compra de un usuario que haya comprado una bicicleta, ofrecer la posibilidad de comprar accesorios adicionales.

Supón que, una vez que el usuario compra una bicicleta, recibe el siguiente documento HTML. Guarda su contenido en un fichero:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Resumen de la compra</title>
    <link rel="stylesheet" href="compra.css" type="text/css">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script src="accesorios.js" type="text/javascript"></script>
    <script src="compra.js" type="text/javascript"></script>
  </head>
  <body>
    <h1>Resumen de la compra</h1>

    <table id="compra">
      <thead>
        <tr>
          <td>Producto</td>
          <td>Unidades</td>
          <td>Precio</td>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td>Bicicleta de montaña 28 v. talla L</td>
          <td>1</td>
          <td>400.00</td>
        </tr>
        <tr id="total">
          <td>Total:</td>
          <td></td>
          <td id="totalval">400.00</td>
        </tr>
      </tbody>
    </table>

    <h2>Accesorios opcionales</h2>
    <form>
      <select id="accesorios">
      </select>
      <input id="cantidad" type="number" value="1" min="1" max="99">
      <input id="boton" type="button" value="Comprar">
    </form>
```

```
</body>
</html>
```

El documento está asociado a una hoja de estilo llamada **compra.css** que se muestra a continuación, que debes guardar en un fichero con dicho nombre en el mismo directorio:

```
#compra {
  min-width: 500px;
}

#compra thead {
  background-color: black;
  color: white;
  font-weight: bold;
}

#compra td {
  padding-left: 0.5ex;
  padding-right: 0.5ex;
}

#total {
  font-weight: bold;
}

#compra td:nth-child(2),
#compra td:nth-child(3) {
  text-align: right;
}

#cantidad {
  text-align: right;
}
```

Además, como puedes observar, incluye dos ficheros JavaScript. El primero, **accesorios.js**, define la lista de accesorios que se pueden comprar, cada uno de ellos con una referencia, etiqueta textual y un precio. Se reproduce el contenido de este fichero a continuación, que debes también guardar con dicho nombre en el mismo directorio:

```
/*
 * Lista de accesorios que se pueden comprar.
 */
var accesorios = {
  rf1x1: {
    etiqueta: "Botellín de agua 0,5L",
    precio: 4.0
  },
  rf1x2: {
    etiqueta: "Botellín de agua 0,9L",
    precio: 5.0
  },
  rf2x1: {
    etiqueta: "Faro delantero",
    precio: 15.5
  },
  rf2x2: {
    etiqueta: "Faro trasero",
    precio: 12
  },
  rf2x3: {
    etiqueta: "Pack faros delantero y trasero",
```

```
} } precio: 23.99  
}
```

El otro fichero, **compra.js**, incluye el código necesario para implementar la funcionalidad de compra de accesorios. De este fichero no se proporcionará el código, sino que debes programarlo siguiendo las instrucciones de los siguientes apartados. Puedes ir creando el fichero, con dicho nombre y en el mismo directorio, dejándolo vacío por el momento.

Es importante que guardes los cuatro ficheros en el mismo directorio para que el navegador los encuentre cuando cargue el fichero HTML.

Ejercicio 2

El menú de accesorios

El control **select** con identificador **accesorios** del documento HTML está, de momento, vacío. Debes rellenar sus opciones mediante JavaScript a partir de la lista de accesorios. El objetivo es que tu programa JavaScript añada automáticamente un elemento **option** por cada uno de los accesorios disponibles, de tal forma que el resultado final sea equivalente a que hubieses introducido los datos a mano de la siguiente forma:

```
<select id="accesorios">  
  <option value="rf1x1">Botellín de agua 0,5L (4.00€)</option>  
  <option value="rf1x2">Botellín de agua 0,9L (5.00€)</option>  
  (...)  
</select>
```

Programa en el fichero **compra.js** el código en una función que rellene las opciones de dicho elemento **select**, y registre dicha función para que se ejecute en cuanto el navegador haya cargado el árbol DOM del documento.

Comprueba que funciona recargando la página en el navegador (presiona la tecla **Shift** al mismo tiempo que presionas el botón de recargar) y viendo si aparecen todos los elementos del catálogo en el menú desplegable.

Se proporcionan a continuación algunas pistas sobre cómo resolver el ejercicio:

- Para acceder desde tu función al objeto que almacena la lista de accesorios que se declara en **accesorios.js** solo tienes que hacer referencia a la variable en cuestión por su nombre: **accesorios**.
- Puedes iterar la lista de accesorios de la siguiente forma:

```
var referencia;  
for (referencia in accesorios) {  
  var accesorio = accesorios[referencia];
```

```
    console.log(accesorio.etiqueta);  
    (...)  
}
```

- Dado un dato de tipo numérico, puedes convertirlo a cadena de texto con dos dígitos decimales llamando a su método **toFixed** con parámetro 2:

```
accesorio.precio.toFixed(2);
```

- Crea cada elemento **option** mediante JQuery, añádele el atributo **value** junto con su valor con el método **attr()** y añádele su contenido textual mediante el método **text()**. Finalmente, añade este elemento como hijo del elemento **select**.
- Si usas Chrome, Firefox o Edge, el navegador incorpora unas herramientas para el desarrollador que te informan de si hay algún error de sintaxis, te permiten ver la consola de JavaScript y te permiten depurar el programa mediante puntos de ruptura y ejecución paso a paso, así como examinar en cada momento qué variables están accesibles y su valor. Te resultarán muy útiles para realizar esta práctica. También podrás ejecutar cualquier instrucciones de JavaScript directamente desde la consola de JavaScript, lo cual te vendrá bien si quieres comprobar si ciertas instrucciones son correctas y se comportan como deseas.

Ejercicio 3

Compra de accesorios (primera aproximación)

Cuando el usuario presione el botón *Comprar*, debe añadirse el accesorio actualmente seleccionado a la tabla de accesorios a comprar. **Programa** en **compra.js** una función que tome el accesorio seleccionado en el menú desplegable y cree una nueva fila al fondo de la tabla, inmediatamente antes de la fila que muestra el importe total. **Registra** la función para que se ejecute cada vez que el usuario presione el botón.

De momento no te preocupes por el número de unidades (asume que se selecciona sólo una) ni actualices el importe total de la compra. Por simplicidad, añade siempre una fila nueva a la tabla, incluso aunque haya ya otras filas para el mismo accesorio en la tabla.

Se proporcionan a continuación algunas pistas:

- Para añadir un manejador de eventos que detecte que presionas el botón, selecciona dicho botón con JQuery y establece el manejador con su método **click()**. Recuerda hacer esto una vez esté cargado el árbol DOM, porque si se

hace antes podría ocurrir que el botón no estuviese creado aún, y por tanto fallase el establecimiento del manejador.

- La función **val()**, aplicada a un control de tipo **select**, devuelve el valor del atributo **value** de la opción que esté seleccionada. Por ejemplo, devuelve la cadena de texto "rf1x1" si se selecciona el botellín de agua de 0,5.
- Utiliza JQuery para crear el elemento **tr** para la nueva fila, crear sus celdas hijas con elementos **td**, añadir el texto de cada una de estas celdas, añadir dichas celdas a la fila y añadir la fila a la tabla.

Ejercicio 4

Compra de accesorios (segunda aproximación)

Mejora la función del apartado anterior para que se tenga en cuenta el número de unidades. La segunda columna debe mostrar el número de unidades seleccionadas y la tercera el precio total de todas las unidades. Además, debes actualizar el precio total de la compra sumando al total anterior el importe de los nuevos accesorios añadidos. Por último, restablece el valor 1 en el control que recoge el número de unidades, para que este sea el número de unidades por defecto si el usuario selecciona algún otro accesorio.

De momento, aunque el accesorio seleccionado esté ya en la tabla, se añade una nueva fila igualmente.

Se proporciona la siguiente pista para resolver este ejercicio:

- La función **parseInt()** de JavaScript devuelve como número entero el valor representado por la cadena de texto que recibe como parámetro. La función **parseFloat()** hace lo mismo para datos de punto flotante. Si recoges un número desde un control de formulario o del contenido de un elemento, lo recibirás como cadena de texto. Debes aplicarle una de estas funciones para convertirlo a tipo número y poder realizar operaciones aritméticas con él.

Ejercicio 5

Compra de accesorios (tercera aproximación)

Mejora la función del apartado anterior para que, si se compra de nuevo un accesorio que ya esté en la tabla, se agreguen las unidades a la fila existente en vez de crear una nueva. Actualiza de forma consistente el precio total de la compra.

Se proporcionan las siguientes pistas para resolver este ejercicio:

- Puede resultar útil establecer un identificador en cada fila de la tabla cuyo valor sea la referencia del accesorio correspondiente. Luego puedes buscar dicha fila, dada una referencia que esté guardada en una variable llamada **referencia**, de la siguiente forma:

```
var tr = $("#" + referencia);
```

- Si necesitas acceder a una columna concreta de una fila, una posible forma es utilizar el método **find** y el selector **:nth-child()**, que devuelve el nodo hijo cuya posición se le indique como parámetro (empezando a contar en 1). Por ejemplo, para seleccionar la segunda columna de una fila dado el identificador de la fila:

```
var tr = $("#" + referencia);  
var tdCantidad = tr.find("td:nth-child(2)");  
var tdPrecio = tr.find("td:nth-child(3)");
```

- Otra alternativa al punto anterior es simplemente seleccionar todos los elementos **td** de la fila y acceder como a un **array**. En este caso, necesitarás convertir los nodos devueltos a nodos de jQuery pasándoselos a la función **\$()**:

```
var tr = $("#" + referencia);  
var listaTd = tr.find("td");  
var tdCantidad = $(listaTd[1]);  
var tdPrecio = $(listaTd[2]);
```

Aplicaciones Web (OpenCourseWare, 2023)

uc3m | Universidad **Carlos III** de Madrid
Departamento de Ingeniería Telemática

