

EJERCICIO 1 - ELEMENTO TABLA HTML

HTML compone tablas de datos utilizando diferentes etiquetas, en un esquema básico obtenido de W3Schools, podemos cómo se enlazan las diferentes etiquetas:

```
<table>
  <tr>
    <th>Company</th>
    <th>Contact</th>
    <th>Country</th>
  </tr>
  <tr>
    <td>Alfreds Futterkiste</td>
    <td>Maria Anders</td>
    <td>Germany</td>
  </tr>
  <tr>
    <td>Centro comercial Moctezuma</td>
    <td>Francisco Chang</td>
    <td>Mexico</td>
  </tr>
</table>
```

Vuestro trabajo consistirá en realizar una función que reciba un array de objetos a mostrar en una tabla y componga el elemento <table>, con todos los datos que se especifiquen en un array que será la cabecera, y añadiendo una botonera con los botones que se indiquen en un array de botones.

Por lo tanto, vuestra función recibirá 3 parámetros:

- Tabla => un array de objetos que se mostrarán en la tabla
- Cabecera => un array de objetos que contendrán la especificación de la cabecera.
- Botonera => un array de objetos que contendrán la especificación de los botones que se propondrán en cada una de las líneas de la tabla.

Se facilita un esquema inicial del código el siguiente archivo que encontraréis junto a este enunciado:



código_inicial.zip

DATOS

Por ejemplo, si tenemos un array de objetos como el siguiente **empleadosObj**, que mostramos como ejemplo, cada objeto será un elemento **línea** de la tabla.

```
const empleadosObj = [
  {
    nombre: "Ander da Silva",
    fechaNacimiento: "2002-08-06T22:00:00.000Z",
    codigo: "A033",
    fechaIncorporacion: "2022-01-23T23:00:00.000Z",
    salario: 16000,
    categoriaProfesional: "A",
    vacacionesPendientes: 24,
    fechaBaja: null,
    historial: [{ operacion: "alta", fecha: "2022-01-23T23:00:00.000Z", categoriaProfesional: "A",
    salario: 16000 }]
  },
  nombre: "Elena Alonso Alonso", fechaNacimiento: "2000-02-04T23:00:00.000Z", codigo: "A028",
  fechaIncorporacion: "2022-01-18T23:00:00.000Z", salario: 16000, categoriaProfesional: "A",
  vacacionesPendientes: 24, fechaBaja: null, historial: [{ operacion: "alta", fecha: "2022-01-
  18T23:00:00.000Z", categoriaProfesional: "A", salario: 16000 }] },
  { nombre: "Estefanía Cerdán Suárez", fechaNacimiento: "2000-01-26T23:00:00.000Z", codigo: "A036",
  fechaIncorporacion: "2022-01-23T23:00:00.000Z", salario: 16000, categoriaProfesional: "A",
  vacacionesPendientes: 24, fechaBaja: null, historial: [{ operacion: "alta", fecha: "2022-01-
  23T23:00:00.000Z", categoriaProfesional: "A", salario: 16000 }] },
  ...
]
```

CABECERA

Cada objeto del array especifica una de las columnas que tendrá la tabla con la cabecera **‘etiqueta’** y la propiedad del objeto que corresponde al dato que irá en la línea **‘elemento’**.

```
const cabecera = [
  { elemento: "codigo", etiqueta: "Código" },
  { elemento: "nombre", etiqueta: "Nombre" },
  { elemento: "fechaNacimiento", etiqueta: "Fecha de nacimiento" },
  { elemento: "fechaIncorporacion", etiqueta: "Fecha de incorporación" },
  { elemento: "categoriaProfesional", etiqueta: "Categoría" },
  { elemento: "vacacionesPendientes", etiqueta: "Vacaciones" }
];
```

BOTONERA

Al final de cada línea se mostrarán tantos botones como elementos se indiquen en este array de objetos. Cada objeto corresponde a un botón cuyo texto será el que indique la propiedad **‘accion’** y la función que se asociará al evento click: **‘eventoClick’**

```
const botonera = [
  { accion: 'Salario' , eventoClick: (e, fila) => mostrarSalario(e,fila) },
  { accion: 'Bloquear' , eventoClick: (e, fila) => bloquear(e, fila) },
  { accion: 'Seleccionar' , eventoClick: (e, fila) => seleccionar(e, fila) }
]
```

FUNCIÓN *generarTablaHTML*

Nuestra función, **generarTablaHTML**, generará un elemento `<table>` que en nuestro ejemplo añadiremos al elemento `body` del document.

```
const tablaHTML = generarTablaHTML(empleadosObj, cabecera, botonera);

document.body.insertAdjacentElement('beforeend', tablaHTML);
```

SALIDA

Mostraremos una tabla con una línea por cada objeto, las cabeceras adecuadas y los botones recogidos en el array de la botonera (salario, bloquear, seleccionar).

Cada uno de estos botones realizará una acción establecida en la función que gestiona cada evento.

Código	Nombre	Fecha de nacimiento	Fecha de incorporación	Categoría	Vacaciones			
A033	Ander da Silva	2002-08-06	2022-01-23	A	24	Salario	Bloquear	Seleccionar
A047	Casilda Alarcos González	2003-08-31	2022-02-08	A	24	Salario	Bloquear	Seleccionar
A034	Cristina Torralba Goñi	2000-12-04	2022-01-23	A	24	Salario	Bloquear	Seleccionar
A028	Elena Alonso Alonso	2000-02-04	2022-01-18	A	24	Salario	Bloquear	Seleccionar
A036	Estefanía Cerdán Suárez	2000-01-26	2022-01-23	A	24	Salario	Bloquear	Seleccionar

Código	Nombre	Fecha de nacimiento	Fecha de incorporación	Categoría	Vacaciones			
A033	Ander da Silva	2002-08-06T22:00:00.000Z	2022-01-23T23:00:00.000Z	A	24	Salario	Bloquear	Seleccionar
A047	Casilda Alarcos González	2003-08-31T22:00:00.000Z	2022-02-08T23:00:00.000Z	A	24	Salario	Bloquear	Seleccionar
A034	Cristina Torralba Goñi	2000-12-04T23:00:00.000Z	2022-01-23T23:00:00.000Z	A	24	Salario	Bloquear	Seleccionar
A028	Elena Alonso Alonso	2000-02-04T23:00:00.000Z	2022-01-18T23:00:00.000Z	A	24	Salario	Bloquear	Seleccionar
A036	Estefanía Cerdán Suárez	2000-01-26T23:00:00.000Z	2022-01-23T23:00:00.000Z	A	24	Salario	Bloquear	Seleccionar
A009	Evelina Santiago Miras	2001-06-05T22:00:00.000Z	2021-06-28T22:00:00.000Z	A	24	Salario	Bloquear	Seleccionar
A043	Felipa Cachón Chacón	2002-08-07T22:00:00.000Z	2022-02-08T23:00:00.000Z	A	24	Salario	Bloquear	Seleccionar
A029	Fermin López Amigo	2003-09-01T22:00:00.000Z	2022-01-20T23:00:00.000Z	A	24	Salario	Bloquear	Seleccionar
A010	Fernando Marrón Fins	2000-12-05T23:00:00.000Z	2021-06-28T22:00:00.000Z	A	24	Salario	Bloquear	Seleccionar
A012	Genaro Zanco Suárez	2000-02-05T23:00:00.000Z	2021-07-13T22:00:00.000Z	A	24	Salario	Bloquear	Seleccionar

Código inicial	0 puntos
----------------	----------

Implementad el código suministrado en el primer apartado.

El código debéis incluirlo en la siguiente función:

```
function generarTablaHTML(tabla, cabeceras, botones) {  
  // Código a desarrollar  
}
```

La especificación de la función como de las funciones asociadas a los eventos de los botones se indican más adelante.

Función generarTablaHTML		4 puntos
<p>Crear dinámicamente un elemento <table>, establecer las siguientes propiedades de estilo: table_layout: auto table_width: 100%</p>		
Cabecera		1
<ol style="list-style-type: none"> 1. Crear un elemento cabecera <tr> 2. Para cada cabecera añadir un elemento <td> al elemento cabecera e incluid el texto correspondiente. Aplicar los siguientes estilos¹: <ol style="list-style-type: none"> a. border: 2px solid b. font_size : 1.2 em c. column_width: auto 3. Añadir el elemento a la Tabla. 		
Líneas		1
<ol style="list-style-type: none"> 1. Crear un elemento fila <tr> 2. Para cada columna añadir un elemento <td> al elemento fila e incluid el texto correspondiente. Aplicar el estilo border: 1px solid. 3. Añadir el elemento a la Tabla. 4. Al objeto fila (en este caso la fila es un objeto empleado), le asignaremos como propiedad elementoFilaHTML², el propio elemento fila generado. 5. Se procederá con la botonera (para cada fila) 		
Botonera		2
<p>En el caso de que se pase el parámetro botones:</p> <ol style="list-style-type: none"> 1. Crearemos un elemento <div> para la botonera, este elemento contendrá todos los botones. 2. Añadiremos al objeto fila, una propiedad botonesHTML con un array vacío, a este elemento le añadiremos cada uno de los botones que se procesen. 3. Para cada uno de los botones, crearemos un elemento <button> y se lo añadiremos a la botonera. 4. Le añadiremos el texto del botón al botón. 5. Si la especificación del botón incluye el evento, se le añadirá su correspondiente listener con la llamada a la función: (e) => boton.eventoClick(e, fila). Esa recibe 2 parámetros, el propio evento y el objeto fila (en este caso el empleado). 6. Añadiremos el botón al array de botonesHTML del objeto fila. 7. Añadimos la propiedad botoneraHTML al objeto fila. 8. Añadimos la botonera al elemento fila creado en el apartado anterior. 		
La función devuelve el elemento tabla HTML generado.		

¹ Los estilos que se aplican mediante la propiedad style de un elemento HTML, en JavaScript se escriben en camelCase, por ejemplo, el estilo font_size, equivale a fontSize. Se omite el guión y la palabra que iría después del guión se inicia con mayúscula.

² El nombre de propiedad indicado es uno propuesto, podéis cambiarlo libremente.

GESTIÓN DE EVENTOS (funciones)	2 puntos
<p>mostrarSalario</p> <p>Al pulsar en el botón Salario, debemos mostrar el salario del empleado durante 3 segundos:</p> <ul style="list-style-type: none"> • Se creará un elemento <td>, se le incluirá el texto de la propiedad salario del empleado. Los estilos serán font_weight: 'bolder', font_size: '2em'. • Se ocultará la botonera de la fila. • Y se añadirá antes de dicha botonera, el elemento creado. • A los tres segundos se mostrará la botonera de la fila y se eliminará el elemento creado. 	1
<p>bloquear</p> <p>Este botón inhabilitará los botones de la fila.</p> <p>Al pulsar en el botón Bloquear, estableceremos la propiedad activo del empleado a false, y deshabilitaremos todos los botones de la fila.</p>	0,5
<p>seleccionar</p> <p>Al pulsar este botón se marcarán o desmarcarán las filas de nuestra tabla. Las líneas marcadas aparecerán en color rojo y cuando se desmarcan, en negro. Utilizaremos una propiedad, seleccionado que deberá establecerse a tal fin.</p>	0,5

EJERCICIO 2 - SORTEO		4 puntos
<p>Se trata de desarrollar una pantalla que facilite el sorteo de un regalo entre una serie de personas.</p> <p>Nuestra aplicación deberá tomar un Array de nombres, que para las pruebas será constante, (se muestra más abajo).</p> <p>La pantalla mostrará el nombre de un sorteo como cabecera y a continuación el botón PREPARAR, y el listado de personas con su número secuencial (empezando en 1) y su nombre.</p> <p>Al pulsar el botón preparar, asignará, para cada persona un número aleatorio para el sorteo (entre 1 y el número total de personas)</p> <p>Una vez asignado a cada persona un número para el sorteo, este se mostrará a continuación del nombre y el botón “preparar” se inhabilitará y se añadirá el botón “sortear”.</p> <p>Al pulsar el botón sortear, se mostrará en un elemento HTML un mensaje “SORTEANDO” durante 3 segundos, y se procederá a generar el número aleatorio que resultará ganador del sorteo.</p> <p>Al finalizar los 3 segundos, el mensaje desaparecerá y se resaltará la persona ganadora en el listado ampliando el tamaño del texto y cambiándole el color. También se inhabilitará el botón sortear.</p>		
Funcionamiento correcto del Sorteo según las especificaciones		2 puntos
Utilización de clases y métodos		1 punto
Grado de generación HTML por código. Si sólo hay estático <html> <headers> y <body> y el resto se genera por código.		1 punto
Si el botón preparar ordena el listado alfabéticamente de forma correcta y muestra el resultado		1 punto extra.
ARRAY PERSONAS		
<pre>const personas = ['Tania Lloret Sonsol', 'Thiago Sintra Sintra', 'Tomás Porras Puerta', 'Violeta Ahmed', 'Yanira Domínguez Campoy', 'Yassine Paez Polito', 'Ana Marín Morán', 'Carolina Domínguez Domínguez', 'Gerardo Serra Clemente', 'Laura Postergado Plinto', 'Lucía Álvarez Álvarez', 'María Rosa Sinpar Parece', 'Noemí García García', 'Omar Terreros Marinos', 'Sebastián Coto Coteló', 'Umer Mers Sian', 'Vanessa Domingo Nieves', 'África Martín Martín', 'Domingo Bono Cinco', 'Javier González Suárez', 'Juan Jonil Juanco', 'Miguel Fernández Morán', 'Paloma Cerco Del Monte', 'Raúl Lomas Lomas', 'Susana Tomás Martínez', 'Abraham Benevente García', 'María Rodríguez Pérez']</pre>		