

Segundo Parcial Mayo 2024

Para evaluar este examen se tendrá en cuenta tanto el correcto funcionamiento de los ejercicios, como la eficiencia de los mismos, por esa razón un ejercicio podrá obtener la máxima nota siempre y cuando tenga en cuenta ambas premisas.

Los ejercicios deben estar resueltos en una página HTML para demostrar su correcto funcionamiento, pudiéndose utilizar la consola, además se debe utilizar una codificación estricta de JavaScript. **Si un ejercicio no está demostrado, el ejercicio se puntuará con un máximo de la mitad de la nota de dicho ejercicio.**

El código HTML de la página no podrá ser modificado salvo para cargar un nuevo JS o por resolver algún posible error.

La página está construida con Bootstrap. Se valorará su uso, aunque no es necesario para superar el ejercicio.

En cada ejercicio se deberá utilizar los métodos de los diferentes objetos del API de JavaScript para su resolución. En el caso de implementar funcionalidad que ya exista en el API, el ejercicio se verá penalizado por dicha circunstancia.

Los ejercicios se podrán resolver tanto utilizando el API estándar que ofrece el intérprete de JavaScript, o también utilizando la librería de jQuery. En la tarea del examen tienes un archivo denominado examenMayo.zip. Este fichero contiene la web sobre la cual deberás realizar el examen. Despliega esta web en el servidor web para poder realizarla.

Por último, no se pueden presentar errores en las pruebas de los ejercicios. Errores de sintaxis o no captura de excepciones implicarán la reducción de la nota o la pérdida total de la puntuación.

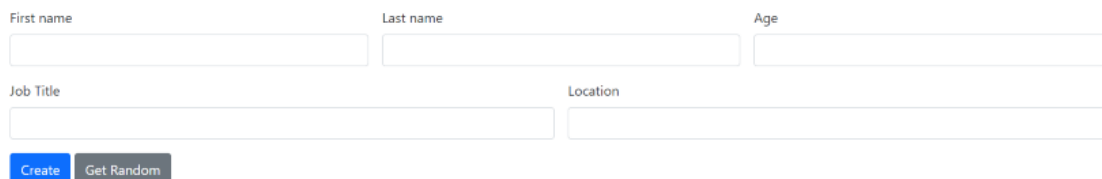
Instrucciones de entrega

Crea una carpeta con tu nombre, ejemplo "LizanoMontalvoPablo", y dentro de ella crea una carpeta por cada ejercicio, ejemplo "ejercicio1", "ejercicio2", etc, las cuales deberán contener la resolución de cada ejercicio. Empaqueta la carpeta principal en un ZIP y súbelo a Delphos.

Si lo ves necesario, explica el ejercicio en un párrafo de la página HTML que hayas creado.

1. DOM creación de elementos (1,5 puntos)

En el index.html de la web proporcionada está dividido en dos partes. Por un lado, tenemos un formulario sobre el cual recogemos una serie de datos de un empleado, y por otro disponemos de una tabla donde mostrar los datos recogidos, la cual dispone de algunos ejemplos. La siguiente imagen muestra el formulario de recogida de datos.



Formulario de recogida de datos:

First name	Last name	Age
<input type="text"/>	<input type="text"/>	<input type="text"/>
Job Title	Location	
<input type="text"/>	<input type="text"/>	
<input type="button" value="Create"/>	<input type="button" value="Get Random"/>	

Imagen 1 Formulario de recogida de datos

Como puedes observar los datos que se solicitan son los de un empleado, siendo **todos obligatorios**. Los datos son:

- Nombre
- Apellido
- Edad
- Puesto de trabajo
- Localización

Por cada empleado deberás **crear una fila en la tabla con dichos datos**. El botón **create**, en lugar de enviar los datos al servidor se limitará a recoger los datos y mostrarlos en la tabla de datos utilizando el API de DOM o jQuery, pero **sin utilizar la propiedad innerHTML**. El botón debe capturar el evento *submit* y cancelar su envío al servidor. Una vez recogido los datos el formulario debe resetearse para comenzar desde cero.

La siguiente imagen muestra la tabla con los datos que se deberán recoger.

Full Name	Age	Job Title	Location	Send	Delete
Vincent Williamson	31	iOS Developer	Washington	<button>Send</button>	<button>Delete</button>
Joseph Smith	27	Project Manager	Somerville, MA	<button>Send</button>	<button>Delete</button>

Imagen 2 Tabla de datos

2. Validación del formulario (2 puntos)

Los datos del formulario deben ser **validados** e indicar al usuario **cuál es el error exacto en el que se ha producido** o mostrar la **retroalimentación positiva**.

La validación debe ser final y en línea, es decir, mostrar la retroalimentación con el error según se cumplimenten los datos.

3. Selección de empleado (2 puntos)

Al clicar en una fila de la tabla, el color de fondo de la fila debe destacarse sobre los demás para indicar que está seleccionada. Los datos de la fila seleccionada deben mostrarse en formato JSON en el textarea de la página.

La selección de otra fila implicará deseleccionar la fila que lo estuviera en primer lugar.

Si cliqueamos por segunda vez en la misma fila, la fila debe quedar deseleccionada y el textarea debe limpiarse.

4. Recogida de datos (1,5 puntos)

En el directorio php dispones de un script denominado **getEmployee.php** al ser invocado devolverá un objeto JSON con los datos de un empleado. El siguiente código muestra los datos obtenidos.

```
{ "firstname": "Joseph", "lastname": "Smith", "age": 27, "jobtitle": "Project Manager", "location": "Somerville, MA" }
```

Al clicar en el botón **Get Random** del formulario deberás recoger este objeto, y a partir de él restaurar con sus datos los campos del formulario utilizando el API de fetch.

5. DOM eliminación de elementos (1,5 puntos)

El botón de **delete** debe **borrar la fila de la tabla seleccionada**.

6. Envío de datos (1,5 puntos)

El botón de **send** dede **recoger los datos seleccionados**, generar un **JSON** con la misma estructura del ejercicio de recogida de datos. En el directorio php dispones de un script **saveEmployee.php**, el cuál espera un JSON en el parámetro **data** en una petición **POST**. El fichero se crea en el directorio **tmp** con el nombre y el apellido del empleado y el contenido del JSON.

Debes mostrar retroalimentación indicando que el empleado ha sido almacenado o si se ha producido un error. Dispones de un modal para hacerlo, aunque puedes utilizar otros medios.