

EXAMEN PRIMERA EVALUACIÓN

DWEC – DISTANCIA CURSO 23-24

FORMA DE ENTREGA:

- Se realizará cada ejercicio dentro de una carpeta diferente: EJ1, EJ2, EJ3.
- Se comprimirá todo en un fichero .zip nombrado siguiendo el formato:
NombreApellido1Apellido2_1ev.zip
- Se subirá el .zip a la Tarea de Examen1ev del moodle.

Claridad y presentación del código, indentación, etc. **(0,5 puntos)**

Ejercicio 1: Registro de Personas (3.0 puntos)

Crea un script en JavaScript llamado "ejercicio1.js" que solicite información de 3 personas, incluyendo fecha de nacimiento, nombre, apellido y número de teléfono. Los datos se almacenarán en arrays distintos. **(0.8 puntos)**

1. Normalización de Datos:

- Los datos introducidos se guardarán en minúsculas, independientemente de cómo hayan sido ingresados. **(0.2 puntos)**

2. Validación de Datos:

- Antes de almacenar la información, se verificará que:
 - Los números de teléfono comiencen con 9, 7 o 6 y tengan exactamente 9 dígitos. **(0.25 puntos)**
 - La fecha siga el formato dd-mm-aaaa. **(0.25 puntos)**
- En caso de incumplimiento de estas reglas, se mostrará un aviso indicando el problema detectado y no se almacenarán los datos en los arrays. **(1.0 punto)**

3. Función "imprimePersonasOrdenadas":

- Implementa una función llamada "imprimePersonasOrdenadas" que imprima en consola los nombres de las personas ingresadas ordenadas alfabéticamente por nombre, con la primera letra en mayúscula. **(0.5 puntos)**
- El formato de salida en consola será: [Nombre]

Ejercicio 2: Clases (3.0 puntos)

1. Clase Persona (1.0 punto):

- Crea una clase llamada `Persona` con las propiedades `nombre`, `apellido` y `fechaNacimiento`.
- Implementa funciones `getNombre`, `getApellido` y `getNacimiento` para obtener los valores de las propiedades.
- Implementa funciones `setNombre`, `setApellido` y `setNacimiento` que no aceptarán datos vacíos.

2. Clase **PersonaCivil** (1.5 puntos):

- Crea una clase llamada `PersonaCivil` que herede de la clase `Persona` y agregue las propiedades `dni` y `telefono`.
- Implementa funciones `getDNI`, `getTelefono`, `setDNI` y `setTelefono`.
- Añade una función `imprimirDatos` que imprima en consola todos los datos de la persona civil siguiendo el patrón indicado:

[nombre][apellido][fechaDeNacimiento][DNI][teléfono]

3. Creación de Personas Civiles (0.5 puntos):

- Crea tres instancias de la clase `PersonaCivil` con todos sus datos.
- Utiliza la función `imprimirDatos` de la clase `PersonaCivil` para imprimir en consola los datos de todas las personas civiles.

Ejercicio 3: Validación y Eventos en Formulario (3.5 puntos)

1. Dado el siguiente formulario:

- Nombre *:
- Primer Apellido *:
- Segundo Apellido:
- Fecha de nacimiento:
- DNI:
- Email *:
- Contraseña *:
- Contraseña (repetición) *:

2. Eventos y Validaciones:

- Implementa un script en JavaScript llamado `validacion.js` y enlázalo con el formulario HTML. **(0.25 puntos)**
- Añade eventos para validar y manipular los datos introducidos en el formulario según los siguientes criterios:
 - **Mayúsculas Automáticas (0.25 puntos):**
 - Una vez que los datos se introduzcan en los campos de texto del formulario, deben convertirse automáticamente a mayúsculas en los campos de "Nombre", "Primer Apellido" y "Segundo Apellido".
 - **Validación de Fecha de Nacimiento:**
 - La fecha de nacimiento debe validarse mediante una expresión regular para garantizar el formato dd/mm/aaaa. **(0.25 puntos)**
 - Verifica que la edad del usuario a la fecha actual sea mayor de 16 años. Esta validación se realizará después de introducir la fecha. **(0.75 puntos)**
 - **Validación de DNI (0.25 puntos):**
 - El DNI debe validarse mediante una expresión regular que permita únicamente 8 números y una letra. Esta validación se realizará después de introducir el DNI.
 - **Validación de Email (0.5 puntos):**

- El email se validará utilizando una expresión regular. Antes de la @, puede contener cualquier carácter alfanumérico y los caracteres . y -. Después de la @, puede contener caracteres alfanuméricos, y el tipo de dominio puede tener una longitud de 2 o 3 caracteres. La validación se realizará después de introducir el email.
- **Coincidencia y Longitud de Contraseñas (0.75 puntos):**
 - Los dos campos de contraseña deben coincidir y tener una longitud mínima de 8 caracteres, al menos una letra minúscula, una letra mayúscula y un dígito. Se mostrará un mensaje indicativo si no se cumplen estas condiciones. La validación se realizará al salir del campo "Contraseña (repetición)".
- **Comprobación de Campos Obligatorios (0.5 puntos):**
 - Al pulsar el botón "Enviar", verifica si los campos marcados con * han sido introducidos. En caso negativo, muestra un mensaje indicativo de los campos no introducidos.

Recuerda estructurar tu código de manera clara y utilizar comentarios para explicar cada sección del código. ¡Buena suerte!