|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mostrando Marca_Txurdinaga_Color_H_SUP_01_01.jpg | Curso / *Kurtsoa* | Fecha / *Data* | Nivel / *Maila* | Eval. /*Ebal.* |
| 2022-23 | 2022/11/18 | 2º | 1ª |
| Módulo / *Modulua* | *Kodea / Código* | *U.Didak /Unid didác* | *tipo* | Calificación/*Kalifikazioa* |
| DWC | 2DW3 | 1, 2, 3, 4, 5, 6 | Prozedurazkoa |  |
| Nombre Alumno/a / Ikaslearen izena | Augusto de la Cámara | | | |

* **Para la resolución de estos problemas se valorará positivamente: la claridad del código, la separación del código Javascript del HTML, el uso de convenciones para nombrar funciones y variables, y el uso de comentarios en el código, así como la corrección de los resultados obtenidos.**
* **El examen vale 10 puntos, si la propuesta del examen se resuelve de forma correcta se le aplica el valor especificado en cada apartado.**
* **Crear cada ejercicio en un directorio diferente con el nombre Ejercicio1, Ejercicio2, Ejercicio3, Ejercicio4. Dentro de cada directorio meter todos los archivos que necesitéis para hacer el ejercicio.**
* **Cuando terminéis subiréis un sólo archivo comprimido, con el nombre EVA1\_Apellidos\_Nombre.**

|  |  |
| --- | --- |
| **Total** | **0,5** |
| Se ha separado en todos los programas el código HTML de JavaScript | 0,25 |
| Se han comentado las funciones de todos los programas | 0,25 |

**EJERCICIO 1 – Objetos**

Crea una clase llamada Cuenta que tendrá los siguientes atributos: titular (que es una persona) y cantidad (puede tener decimales). El titular será obligatorio y la cantidad es opcional.

Construye los siguientes métodos para la clase:

1. Un constructor, donde los datos pueden estar vacíos.
2. Los setters y getters para cada uno de los atributos. El atributo cantidad no se puede modificar directamente, sólo ingresando o retirando dinero.
3. mostrar(): Muestra los datos de la cuenta.
4. ingresar(cantidad): se ingresa una cantidad a la cuenta, si la cantidad introducida es negativa, no se hará nada.
5. retirar(cantidad): se retira una cantidad a la cuenta. La cuenta NO puede estar en números rojos.

Una vez definido el objeto.

1. Crea dos nuevas cuentas
2. Meterlas en un array
3. Hacer algún ingreso y retirada
4. Mostrar los resultados

|  |  |
| --- | --- |
| **Total ejercicio** | **3** |
| 1-El objeto ha sido definido correctamente con sus get y set | 1 |
| 1-El método constructor ha sido definido correctamente | 0,5 |
| 1-El método mostrar ha sido definido correctamente | 0,25 |
| 1-El método ingresar ha sido definido correctamente | 0,25 |
| 1-El método retirar ha sido definido correctamente | 0,25 |
| 1-Las 2 cuentas han sido definidas correctamente y almacenadas en el array. | 0,25 |
| 1-Los métodos funcionan correctamente con las cuentas creadas en el array | 0,5 |

**EJERCICIO 2 – Formularios y DOM**

Dispones de una estructura de carpetas llamada ejercicio2 que contiene un archivo HTML, y varias carpetas (css, img y js). Únicamente deberás completar el archivo llamado invierno.js en la carpeta js, de modo que quedará automáticamente referenciado desde el HTML. Únicamente deberás tocar ese archivo. El resto (HTML, css e img) deberán quedar tal y como están.

Deberás crear, como mínimo, las siguientes funciones en dicho archivo:

|  |  |
| --- | --- |
| Función | Descripción |
| crear | Crea una fila en la tabla “participantes” que contendrá las siguientes celdas:  - Nombre: guardará el texto del input “nombre”.  - Puntuación: guardará el texto del input “puntuación”.  - Pais: mostrará una imagen que se obtendrá de la carpeta img, y a continuación el nombre del país. Es decir, si introducimos el país “Alemania”, deberá mostrarse la imagen Alemania.png que está en la carpeta img, seguida de la palabra “Alemania”. |
| borrarFormulario | Elimina la información del formulario. Este método se ejecuta al final de la función crear. |
| borrarUltimo | Elimina la última fila de la tabla. |
| validar | Deberá validar el formulario antes de crear la fila de la tabla teniendo en cuenta:   * Los tres campos deben ser obligatorios. * El campo de nombre solo puede contener caracteres alfabéticos (letras mayúsculas y minúsculas, letras vocales acentuadas, letra ñ y espacios). * El campo nota debe ser un número de 0 a 100 (incluidos). * El campo país deberá tener la misma valoración que el nombre.   Al salir de cada elemento del formulario, en caso de que no se cumpla la validación, mostrará un error en un alert indicando cuál ha sido el problema.  \*Se tendrá en cuenta que no se duplican los métodos para validar el nombre y el país y que se indica al usuario cuál de los dos inputs se está validando. |
| colorear | Al pasar el ratón por encima de las filas de la tabla, estas se colorean de color rojo. |
| decolorear | Al salir el ratón de una fila, vuelve a recuperar el color inicial (ninguno). |

|  |  |
| --- | --- |
| **Total ejercicio** | **3,5** |
| El método crear muestra los tres valores y la imagen (si la hay) (0,25 por elemento). | 1 |
| El método borrarUltimo borra la última fila de la tabla. | 0,5 |
| El método borrar elimina la información del formulario correctamente | 0,1 |
| Se validan los campos requeridos. | 0,15 |
| Se valida el campo numérico. | 0,15 |
| Se validan correctamente la expresión regular del texto. | 0,3 |
| Solo se utiliza un método para validar el país y el nombre. | 0,3 |
| El método que valida el país y el nombre muestra el nombre del campo que da error. | 0,25 |
| El método colorear cambia el fondo de toda la fila. | 0,25 |
| El método decolorear vuelve a poner la fila del color inicial. | 0,25 |
| Los métodos colorear y decolorear se aplican correctamente a las celdas creadas. | 0,25 |

**EJERCICIO 3 – Correcciones**

Te han dado para corregir un pequeño código de la aplicación en la que trabajas. El código está compuesto por un archivo HTML, un archivo css y un archivo js (carpeta Ejercicio3). El código que **tiene errores y no funciona** debe de cargar inicialmente un formulario con tres campos para introducir por el usuario. Al pulsar el botón *Validar* debe de validar los 3 campos (los tres son obligatorios y el campo nombre debe de contener únicamente letras y espacios). Cuando se pulsa en el botón, todos los errores que se produzcan se deben de mostrar en el campo habilitado para tal fin.

En caso de estar correctos todos los campos, se mostrará la parte inferior del formulario en el que al seleccionar en el desplegable una actividad se mostrará una información: un error si es una actividad de riesgo y la edad es menor de 18; o un agradecimiento en cualquier otro caso. Al cambiar de actividad, el mensaje se debe actualizar. El botón de *reset* debe de dejar el formulario como al principio para poder ser usado de nuevo, esta funcionalidad hay que implementarla siguiendo la lógica del resto del programa.

Para corregir el código *únicamente* se puede modificar el archivo js. Los errores pueden ser de escritura, lógica u orden.

|  |  |
| --- | --- |
| **Total ejercicio** | **3** |
| Corregir los errores del código (0,25 por elemento). | 2,5 |
| Añadir la funcionalidad del botón reset. | 0,5 |