

Tarea para DWECC04.

Caso práctico



Elaboración propia.

Antonio ha avanzado mucho en su estudio de JavaScript tras investigar las características de los objetos predefinidos en JavaScript. Es el momento de ver cómo puede organizar los datos que genera su aplicación de JavaScript, de forma que le permita gestionarlos eficientemente. Para ello **Juan** le explica cómo podrá crear nuevos objetos y asignarles propiedades y métodos en JavaScript.

Antonio se pone manos a la hora con esta nueva sección y realiza pequeñas aplicaciones JavaScript usando dichos objetos.

1.- Contexto.

¿Qué relación tiene esta tarea con los contenidos y con las demás tareas y unidades?

CONTEXTO DE LA TAREA DENTRO DEL MÓDULO

Con esta tarea continuamos realizando pequeñas aplicaciones en JavaScript. En este caso crearemos nuevos objetos con sus respectivas propiedades y métodos, además del uso de arrays.

Esta tarea nos permitirá un mejor seguimiento del módulo a nivel práctico, así como la realización de las tareas de unidades posteriores.

2.- Enunciado.

¿Qué te pedimos que hagas?

ENUNCIADO DE LA TAREA.

Vamos a realizar una serie de tareas sencillas:

1. Vamos a crear un **array** donde aparezcan los primeros 15 número primos de forma descendente. O sea, de mayor a menor. (13,11,7,5,3,2,1). Además, deberéis introducir los elementos de tal forma que no se tengan que ordenar estos números. O sea, que deberéis meter el 1; después sería (2,1) y a continuación (3,2,1) y etc.

1. Además crearéis una función que se llame **invertirArray(arrayInvertir)** que lo que haga sea invertir el **array** en un nuevo **array** que será el valor que se devuelva con **return**. Está prohibido utilizar esta función para introducir los elementos en el momento de la creación. Para llamarla podemos utilizar:

```
var nuevoArrayInvertido = InvertirArray(arrayPrimos);
```

2. Crear una función llamada **VisualizarArray(arrayVisualizar)** que muestre cualquier **array** de números.
2. Cree una clase llamada **Identidad**. Esta clase tendrá los siguientes métodos:
 1. El constructor será nulo, ya que esta clase es más bien una agrupación de métodos relacionados y no una clase auténtica.
 2. **GenerarLetra** que recibe como entrada un DNI sin letra y obtienes una cadena DNI con esa letra.
 3. **ComprobarDNI**. Comprueba si el número del DNI es correcto o no lo es. Devuelve **true** si es correcto y devuelve **false** si no lo es.

Nota: El DNI se considera que siempre debe tener letra. La letra es un dígito de control. Podéis buscar el algoritmo en: https://es.wikipedia.org/wiki/N%C3%BAmero_de_identificaci%C3%B3n_fiscal

3. Vamos a crear una clase llamada **Diputado**. Con varios diputados haremos un **array** llamado **arrayParlamento**.
 1. La clase Diputado va a estar compuesta por las siguientes propiedades.
 1. Nombre. El nombre de pila.
 2. Apellidos. Los apellidos.

3. DNI. El DNI. Debe ser correcto y para comprobarlo
4. Edad. Debe ser mayor que 0 y menor de 200.
5. Partido. Siglas del partido al que pertenece.
2. El constructor deberá ser algo similar a: **Diputado (nombre, apellidos, dni, edad, partido)**
3. Diputado debe disponer de diversos métodos. Unos serán get que devolverán el valor indicado, y otros serán set que establecerán un nuevo valor en la propiedad a la que se refieren. Estos serán los métodos:
 1. **getNombre()** -> Devuelve el nombre
 2. **setNombre(nuevoNombre)**
 3. **getApellidos()** -> Devuelve los apellidos.
 4. **setApellidos(nuevosApellidos)**
 5. **getDNI()**
 6. **setDNI(NuevoDNI)** -> Devuelve false si no se ha podido cambiar, true si se ha podido cambiar porque no es correcto ese DNI)
 7. **getEdad()** -> Devuelve la edad.
 8. **setEdad()**
 9. **setPartido()**
 10. **getPartido()** -> Devuelve el partido.
 11. **toString()** -> Mostrar todos las propiedades de Diputado.
4. Añada los siguientes elementos al array y ordénelos por edad.

"Herminio","Cabezas",12349,30,"Verde"
"Manolo","López",12349,20,"Azul"
"Jacobo","Sánchez",12349,53,"Amarillo"

3.- Criterios de corrección y puntuación.

¿Cómo valoramos y puntuamos tu tarea?

CRITERIOS DE CORRECCIÓN Y PUNTUACIÓN DE LA TAREA.

Los puntos que tiene esta tarea están asignados de la siguiente forma:

- Ejercicio 1. 2,5 puntos.
- Ejercicio 2. 2 puntos.
- Ejercicio 3. 4,5 puntos.

- Claridad de código siguiendo las reglas indicadas en el foro (ya están colgadas en el foro general) y en los comentarios. Esta vez se corregirá de forma mucho más severa. 1 punto.

Además, las faltas de ortografía si son acentos o vulgarismos quitan 0,05 menos por cada falta. Si son faltas normales se quita 0,1 por cada falta. No seguir las especificaciones, salvo que se indique otra cosa en los foros, podría llegar a anular un apartado o un ejercicio completamente (dependerá del error).

Total: 10 puntos máximo.

4.- Indicaciones de entrega.

¿Cómo debes entregar la tarea para su corrección?

INDICACIONES DE ENTREGA PARA LA TAREA.

Una vez realizada la tarea elaborarás un único documento donde figuren las respuestas correspondientes. El envío se realizará a través de la plataforma de la forma establecida para ello, y el archivo se nombrará siguiendo las siguientes pautas:

EjercicioX.html

Asegúrate que el nombre no contenga la letra ñ, tildes ni caracteres especiales extraños. Así por ejemplo la alumna **Begoña Sánchez Mañas para la cuarta unidad del MP de DWEC**, debería nombrar esta tarea como...

Sanchez_Manas_Begona_DWEC04_Tarea.zip