

Arrays Functions

- 1.- Implementa una función que, dados dos parámetros: un array y un elemento, devuelva Verdadero si ese elemento está en el array o Falso en caso contrario. *Nota: ya existe en JS una función que hace algo parecido. No quiero que uses esa función, quiero que implementes tú la tuya.*
- 2.- Implementa una función que, dada una cadena con palabras separadas por espacio, muestre por consola la longitud de cada palabra del array.
- 3.- Crea una función que, dado un array de números, devuelva otro array...
 - a) ... en el que en cada celda tiene la raíz cuadrada del elemento de esa celda del array original.
 - b) ... en el que cada celda contiene el resultado de elevar el elemento de la celda a su índice dentro del array.
 - c) ... que solo contenga los números pares.
- 3.- Implementa una función que, dado un array de cadenas, cambie las palabras de 4 y 6 caracteres por tantos asteriscos como longitud tenga esa palabra. *Ejemplo: cambia hola por ****. Cambia coraza por ******
- 4.- Sin hacer uso de bucles FOR o WHILE, implementa una forma de mostrar por consola los elementos de un array de cadenas con la siguiente plantilla: *En la celda __ tenemos el valor __*

Objetos

1. Se pretende crear el objeto ALUMNO que almacenará los datos de cada alumno del instituto. Para ello es necesario ir creando las siguientes estructuras de objetos:
 - TUTOR: el cual tendrá los campos: nombre, edad, DNI y título universitario.
 - ASIGNATURA. De cada asignatura necesitamos saber su nombre, el curso al que pertenece (un número entero) y el número de horas totales.

A continuación crea una lista de, al menos, 4 objetos asignaturas diferentes y crea también dos tutores.

Haciendo uso de los objetos TUTOR y de la lista de objetos ASIGNATURA, define un objeto ALUMNO:

Un alumno tendrá un nombre, una edad, el ciclo que estudia, el curso en el que está (un número entero), un tutor, una lista de asignaturas y una lista de nota media de cada asignatura.

Crea un objeto ALUMNO con todo lo indicado.

2.- Haciendo uso de las distintas herramientas, recorre el objeto ALUMNO y muestra, dentro de un DIV maquettato en CSS, todos los datos del alumno.

Imagina que estás haciendo una aplicación para gestionar alumnos que va a usar gente que no sabe programar. Es decir, muestra la información del alumno de la forma más clara posible.

3. Añade las siguientes funciones al objeto TUTOR:

- `mostrar()` : devuelve una cadena con todos los datos del objeto (intenta que la información devuelta sea clara).
- `cambiarNombre(nuevo)` : permite cambiar el nombre del TUTOR.

4. Añade las siguientes funciones al objeto ASIGNATURA:

- `mostrar()` : devuelve una cadena con todos los datos del objeto (intenta que la información devuelta sea clara).
- `cambiarHoras(nueva)` : permite cambiar el número de horas del objeto.

5. Añade las siguientes funciones al objeto ALUMNO:

- `calcularMedia()` : en base a la lista de medias del alumno, devuelve la media total del curso de ese alumno.
- `MediaAsignatura()` : devuelve una cadena con el nombre de cada asignatura y, a continuación, su media.
- `mostrar()` : devuelve una cadena con todos los datos del objeto (intenta que la información devuelta sea clara). Incluye las asignaturas con sus medias y la media total calculada en los puntos anteriores.

Nota: sería interesante que practicaras el crear métodos para mostrar/cambiar campos del objeto ALUMNO.

6. Haciendo uso de las funciones implementadas en los ejercicios anteriores (y de alguna forma para recorrer objetos) recorre el objeto ALUMNO y muestra, dentro de un DIV maquettato en CSS, todos los datos del alumno.

Imagina que estás haciendo una aplicación para gestionar alumnos que va a usar gente que no sabe programar. Es decir, muestra la información del alumno de la forma más clara posible.


7. La empresa donde estás realizando las prácticas te han pedido que diseñes parte de la API de datos que van a servir en su web. Piden concretamente lo siguiente:

El objeto USUARIO el cual tendrá los campos:

- **genre:** indica el genero del usuario (masculino o femenino)
- **datos:** este campo guarda la información personal del usuario. Campos:
 - **name:** nombre del usuario.
 - **lastname:** apellidos del usuario
 - **age:** numero entero que representa la edad.
 - **Dni:** cadena alfanumérica que almacena el documento nacional de identidad del usuario.
- **address:** sirve para almacenar información de la dirección del usuario. Se compone de los campos:
 - **street**
 - **number**
 - **city**
 - **country**
- **coordinates:** indica las coordenadas terrestres del lugar actual del usuario. Se compone de los campos: **latitude** y **longitude**
- **login:** guarda los datos de registro del usuario. Campos:
 - **id:** cadena alfanumérica que almacena la ID de conexión.
 - **User:** nombre de usuario.
 - **Pass:** clave de usuario.
 - **Picture:** ruta con el archivo de la imagen del avatar del usuario.
- **contact:** almacena la información referente a las posibles formas de ponerse en contacto con el usuario. Campos:
 - **phone:** teléfono del usuario (solo números)
 - **email**
 - **SSNN:** lista con las direcciones de los perfiles de redes sociales del usuario (al menos 3)
- **pictures:** guarda diferentes fotos reales del usuario. Campos:
 - **big:** foto del usuario en tamaño grande (500x500)
 - **small:** foto del usuario en tamaño pequeño (100x100)
- **softwareSkills:** guarda una lista con el nombre del software que el usuario domina (al menos 5)

Crea un objeto con la estructura indicada y rellénalo con valores. **Nota:** busca imágenes, metelas en la carpeta *img* junto al archivo Javascript y coloca la ruta correcta al crear el objeto para colocar.

8. Usando el USUARIO creado en el ejercicio anterior muestra por pantalla la siguiente interfaz:



Nombre: Prueba

Apellido: Prueba

Usuario: pruebaPrueba

Edad: 25

Dirección: C/ Prueba

Habilidades: **Databases** **Graphic Design** **Computer Programming** **Presentation Tools** **Social Media**

Teléfono: 666666666

Email: prueba@gmail.com

Control de Eventos

1.- Crea un documento HTML con un section que en su interior tiene un párrafo y este a su vez, tiene un enlace al campus del instituto. (Maqueta los elementos para que se distingan unos de otros).

Todos los elementos mencionados tienen una función asociada al evento click del ratón. El section cambiará su color de fondo, el párrafo cambiará su anchura y el enlace mostrará un panel `confirm` para decirle al usuario si se mueve a la página enlazada o no (es decir, al pulsar el enlace este no va a la pagina enlazada a no ser que lo diga el usuario).

- Prueba la fase de burbuja y la fase de captura ¿ves la diferencia?
- Con todas las funciones de evento definidas y asignadas: haz que solo se ejecute la del enlace.
- Con todas las funciones de evento definidas y asignadas: haz que solo se ejecute la del section.

2.- Crea un documento HTML con dos DIV. Crea un script que, cada vez que se mueva el ratón, indique las posiciones de este de la forma (posx,posy). Usa el primer div para mostrar esa información.

Además, cuando se pulse alguno de los botones del ratón, en el segundo DIV se debe mostrar qué botón se ha pulsado (Izquierdo , derecho o central).

También debe impedir que aparezca el menú contextual al pulsar el botón derecho del ratón.

3.- Crea un documento HTML que tenga de un título h1 con el texto <<Pulsa una tecla>>, y un DIV central de 300x300px maquetado para que se vea bien (borde, tamaño de letra alto, color de fondo...) Utiliza el evento keypress para determinar qué carácter ha escrito el usuario utilizando el teclado y muestralo dentro del div.

4. Crea un documento HTML que conste de un título h1 con el texto <<Usa los cursores>>, y un DIV de tamaño 100x100px (maquetado con borde color negro y color de fondo rojo). El DIV debe aparecer en una posición aleatoria del documento cada vez que se recarga la página.

Utiliza el evento keypress para determinar si el usuario pulsa una tecla del cursor, y en caso afirmativo, utilizando CSS haz que el DIV se desplace 20px en la dirección de flecha elegida por el usuario. Evita que se salga de los bordes del documento.

b) Ahora, en vez de hacer que el DIV se mueva 20px en una de las direcciones al pulsar la tecla correspondiente, el DIV se debe mover en la dirección indicada de 20 en 20 px y no parar hasta que encuentre un borde o hasta que se cambie el sentido de la dirección al pulsar otra tecla de los cursores.

Formularios

1.- Crea un formulario con dos campos de texto y un botón llamado *copiar*. El segundo campo de texto debe estar colocado en modo de solo lectura.

Desarrolla un programa que, cuando se pulse el botón *copiar*, copie todo el contenido del primer campo de texto en el segundo.

2.- Crea un formulario con dos campos de texto. El segundo campo de texto debe estar colocado en modo de solo lectura.

Desarrolla un programa que, conforme se vaya escribiendo en el primer campo, se va a ir copiando ese contenido en el segundo campo. *Nota: busca el evento adecuado para este ejercicio.*

3.- Crea un formulario con varios checkbox (pej, aficiones), un textarea y un botón. Implementa un programa que, tras pulsar el botón se muestre dentro del textarea todas las opciones marcadas.

b) Mejora el apartado anterior quitando el botón. Ahora cada vez que se marque un checkbox, se añadirá al contenido del textarea. Si se desmarca, hay que quitarlo.

4.- Realiza un programa que, haciendo uso de dos campos de texto y un botón, realice la conversión de dólares y euros ($1\$ = 0.83\text{€}$).

Completa el ejercicio añadiendo dos campos y un botón nuevos para implementar la conversión contraria (de euros a dólares).

b) Mejora el ejercicio anterior usando solo dos campos y un botón para realizar ambas conversiones. Para saber qué conversión debemos realizar en cada momento, esta se debe seleccionar de una lista desplegable.

5.- Crea un formulario con dos campos de texto y un botón llamado COMPROBAR. El botón debe comprobar si el contenido de uno de los campos es anagrama del otro. Mostrará el resultado haciendo uso de la función alert. Si ambos campos están vacíos, no se considerará que son anagramas.

Nota: una palabra es anagrama de otra si las dos tienen las mismas letras, con el mismo número de apariciones, pero en un orden diferente.

6.- Crea un formulario con varios elementos textbox (pej, países) y una lista desplegable vacía.

Implementa el código necesario para que, cada vez que se marque un checkbox, se cree una entrada nueva en la lista desplegable con el país correspondiente. Si se desmarca un checkbox, debe eliminarse el país correspondiente de la lista.

7.- Crea un textarea y un botón llamado *Publicar*. Cada vez que se pulse el botón *Publicar*, se va a crear un div cuyo contenido será el texto del textarea. *Nota: maqueta de manera vistosa los DIV creados.*

- b) Haz que sólo se publiquen en el div los 50 primeros caracteres escritos en el textarea.
- c) Limita a que sólo se puedan escribir 100 caracteres en el textarea. Después de los 100 caracteres escritos, no se puede seguir escribiendo. El botón Publicar no debe cambiar su comportamiento. *Nota: busca el evento adecuado para este ejercicio.*

8.- Coloca un div que ocupe toda la pantalla, que tenga una altura de 200px y un color de fondo (el que quieras). Crea un programa que permita cambiar el color de ese DIV por otros usando:

- a) Indicando el color deseado entre alguna de las opciones de un bloque de botones radio (*al menos 3 colores para elegir*).
- b) Indicando el color deseado entre los elementos de una lista desplegable (*al menos 3 colores para elegir*).

9.- Crea un campo de texto en el que se deba colocar un DNI (nueve dígitos y una letra). Cuando ese elemento pierda el foco (crea más elementos para que eso ocurra), se debe validar lo siguiente:

- Que hay nueve números y una letra.
- Que la letra introducida es la letra correcta (*Aquí se indica: <http://www.interior.gob.es/web/servicios-al-ciudadano/dni/calculo-del-digito-de-control-del-nif-nie>*)

10.- Crea un formulario con dos listas desplegables y un botón. La primera lista desplegable tendrá los valores: vacío (el valor por defecto), Intel, AMD y Nvidia.

La segunda lista inicialmente no tendrá ningún elemento. Al elegir un valor en la segunda, la segunda lista se llenará con los valores siguientes:

- Para Intel: HD 3000, HD 4000, IRIS 600.
- Para AMD: RX Series 500, Vega Series, RX Series 6000.
- Para Nvidia: GTX Serie 1000, GTX Serie 2000, GTX Serie 3000.

Además, al pulsar el botón, este debe comprobar que ambas listas tienen un valor seleccionado e informará usando un alert de si eso ocurre o no.

Expresiones Regulares

1.- ¿Qué validan las siguientes expresiones regulares?

- a) `ho*a`
- b) `([B-D] | [F-H] | [J-N] | [P-T] | [V-Z])`
- c) `^[0-9]{5}$`
- d) `[0-4][0-9][0-2][0-9]{3}`
- e) `([0-4][0-9]|5[0-2])[0-9]{3}`
- f) `(^.[a-z])(.*[A-Z])(.*\d)[a-zA-Z]{8,12}$`

2.- Sea la expresión regular: `/^[A-J]\w+\ses\s\w+/` Indica qué cadenas (o parte de ellas) cumplen el patrón indicado.

- a) Jaime no es guapo.
- b) Jose es un llorón.
- c) Guille deja de hablar.
- d) David es muy divertido.
- e) Lorena es pizpireta.
- f) Sergio es listo.
- g) Ivan es callado.

3.- ¿Qué validan las siguientes expresiones regulares?

- a) `\d{6}$`
- b) `^[a-z0-9-]+$`
- c) `^95[0-9]\s*(\d{6}$)`
- d) `(\d{2}\s){2}`
- e) `(\d{2}\s){2}\d{2}$`
- f) `^95[0-9]\s*(\d{2}\s){2}\d{2}$`
- g) `(\d{3}\s){2}`
- h) `^95[0-9]\s*(\d{3}\s){2}`
- i) `95[0-9]\s* ((\d{6}$) | ((\d{2}\s){2}\d{2}$))`
- j) `95[0-9]\s* ((\d{6}$) | ((\d{2}\s){2}\d{2}$) | ((\d{3}\s){2}))`

4.- Crea un script que valide un DNI español usando expresiones regulares. El DNI está compuesto de 8 dígitos entre el cero y el nueve y una letra mayúscula.

5.- Indica cual sería la expresión regular necesaria para validar un email. Recuerda:

- El email debe de empezar por una cadena de caracteres que no empiecen por @ o espacio en blanco. La longitud de esta cadena debe ser mayor a cero.
- A continuación, debe aparecer el símbolo @.
- Después, una cadena de caracteres que no empiecen por @ (arroba), "." o espacio en blanco, cuya longitud sea mayor a cero.
- Esa cadena debe estar seguida de un "." (punto)
- Después, otra vez, una cadena de caracteres que no empiecen por @, "." o espacio en blanco, cuya longitud sea mayor a cero.

6.- Indica cuáles serían las expresiones regulares que hay que usar para validar las siguientes entradas de un formulario:

- a) Nombre de usuario: debe empezar por letra (mayúscula o minúscula). Puede tener números y los caracteres guion medio y guion bajo (ningún otro). No puede contener espacios. Longitud máxima de 16 caracteres.
- b) Contraseña: longitud mínima de 8 caracteres y máxima de 18. Puede contener números, espacios y cualquier carácter. Debe contener alguna mayúscula.
- c) Valor hexadecimal: debe comenzar por el carácter almohadilla # o 3 caracteres hexadecimales válidos: letras de la A a la F (mayúsculas o minúsculas) y números del 0 al 9. Nada de espacios o algún otro carácter.
- d) URL: comienza por `http://` o `https://`. A continuación puede aparecer o no la cadena `www`. Después irá el nombre del dominio (números, letras y guion medio). Punto. Puede aparecer o no el subdominio (lo mismo que el dominio). Punto. Y finaliza con la extensión (2 o tres letras. No números ni caracteres). La extensión puede repetirse.
- e) Una IP válida: cuatro bloques de dígitos separados por el carácter punto (el último bloque no lleva el punto). En cada bloque puede aparecer 1, 2 o 3 dígitos. Ninguna letra o carácter están permitidos en los bloques.
 - Mejora el apartado anterior para que en los bloques solo puedan aparecer números entre el 0 y el 255.
- f) Etiqueta HTML: comienza por el signo `<`. A continuación lleva letras mayúsculas o minúsculas (nada de números o caracteres). Finaliza con el carácter `/` (puede aparecer o no) y con el carácter `>`.
 - Mejora el apartado anterior para que pueda reconocer atributos: `atributo="valor"`. Recuerda que se pueden poner varios atributos separados por espacio.