

SECCION 4 (Strings // Numbers)

Practica 23. (23_string.html) -> Método ['strings()']

Crear una página con dos elementos tipo `<input>` para introducir el nombre y el apellido de forma separada, y un botón llamado '*Mostrar Datos*' que al ser pulsado llame al método '*strings()*' que capturará el nombre y el apellido. Una vez se hayan capturado los valores y almacenados en variables, la información que se deberá mostrar por pantalla es la que se muestra en el resultado.

- Los valores almacenados se guardarán en variables independientes.
- El alumno/a utilizará el número de variables como considere necesario.
- Desde la función '*strings()*' se cargarán los diferentes valores en el `<div>` con ['*innerHTML*'].
- Los métodos y propiedades que se tienen que utilizar en la práctica se muestran a la derecha de cada uno de los campos que se introducirán en el elemento `<div>`

*****Resultado*****

Nombre:

Apellido:

Mostrar

Nombre: Homer	[var nombre]
Apellido: Simpson	[var apellido]
Nombre y apellido: Homer Simpson	[concat()]
Tamaño texto: 13	[length]
Mayúsculas: HOMER SIMPSON	[toUpperCase()]
Minúsculas: homer simpson	[toLowerCase()]
Posición del espacio: 6	[search()]
Finaliza con letra 'a': False	[endsWith()]
Comienza por letra 'H': True	[startsWith()]
Extraer solo apellido: Simpson	[slice(), search()]

Practica 24. (24_numbers.html)

Crear una página con dos elementos de formulario que solicite el precio y el descuento de un producto. El resultado final mostrará el precio final del producto con 4 decimales [*toFixed()*].

La función principal se llamará *'capturar()'* y será la encargada de recuperar los valores de los *<input>* y comprobar que realmente son valores numéricos. Esta misma función invocará a la función *'preciofinal()'* que recibirá como parámetros los dos números y tras realizar los cálculos necesarios devolverá el precio final del producto mediante *'return'*.

- Si alguno de los valores introducidos no es numérico se debe informar mediante un mensaje [*alert()*] que los datos introducidos no son válidos y borrará los valores introducidos dentro de los inputs.
- El botón *'Calcular'* será el encargado de invocar a la función *'capturar()'* que recuperará los valores introducidos en el *<input>* para poder realizar los cálculos necesarios.

*****Resultado*****

Precio compra:

Precio Final:

Descuento:

Calcular

SECCION 5 (Arrays)

Practica 25. (25_arrays_length.html) -> Método ['mostrar_array()']

Crear una página que contenga un elemento `<input>` y un botón llamado '*Añadir valor*' que al ser pulsado almacene valores en un array. Posteriormente mediante otro botón llamado '*Mostrar resultado*' se escribirá por pantalla los valores del array que han sido previamente introducidos con el total de números que contiene el array [*length*].

- Los botones tendrán asociado un elemento Listener para invocar a los métodos.
- Los valores del array pueden ser números o textos y se añaden a través del método '*push()*'.

*****Resultado *****

AÑADIR ELEMENTOS

Valor introducido:

Añadir valor

Mostrar

Valores del array: Pedro, Ana, 15, true, 4, hola, Javascript
Tamaño del array: 7

Practica 26. (26_arrays_localizar.html) -> Método ['localizar_array()']

Disponemos del siguiente array:

```
var marcas = ["Audi", "BMW", "Nissan", "Opel", "Jaguar", "Seat", "Fiat", "Tesla"]
```

La web tendrá un botón llamado 'Consultar' que al ser pulsado ejecutará una función que nos solicitará por pantalla el nombre de una marca de un vehículo. Si el nombre del vehículo se encuentra dentro del array ['indexOf()'] lo mostrará por pantalla diciendo 'Vehículo X localizado' y posteriormente será eliminado del array ['splice()']. Si no localiza el valor introducido mostrará el mensaje 'Vehículo X NO localizado' informando por pantalla.

- El array se copiará en la zona superior del archivo .js como una variable global.
- Cuando el array se quede sin ningún elemento en su interior se mostrará el mensaje de "No hay más elementos en el array".

*****Resultado*****

BUSQUEDA ARRAYS

Vehículo SEAT localizado

Consultar

Practica 27. (27_arrays_numeros.html) -> Método ['numeros_array()']

Disponemos del siguiente array:

```
var nums = [7, 8, 2, 8, 15, 13, 50, 16, 2, 11, 19, 24, 36, 14, 21, 43, 17, 22, 41]
```

La web ejecutará un procedimiento al pulsar el botón 'Consultar' que evaluará el array y determinará cual es el número más grande y cuál es el más pequeño. También mostrará el promedio de todos los números con 2 decimales ['toFixed()'].

Aunque el ejercicio se podría realizar a través de los métodos ['Math.max()'] y ['Math.min()'] que nos proporciona el valor máximo y mínimo de un conjunto de valores, es preferible realizar el ejercicio utilizando la sentencia 'for...in' para recorrer arrays.

- El array se copiará en la zona superior del archivo .js como una variable global.
- El promedio se debe realizar después de obtener la suma de todos los valores numéricos que se encuentran dentro del array.
- El texto resultante se mostrará
- Mediante el botón 'Calcular' invocaremos a la función del ejercicio que ejecuta el cálculo. El botón además estará asociado a un evento Listener para su ejecución.

*****Resultado*****

RECORRER ARRAYS

Valor Max: ----

Valor Min: ----

Promedio: ----

Consultar

Practica 28. (28_arrays_par_impar.html) -> Método ['par_impar()']

Crear una página web con un botón llamado 'Añadir número' y un elemento de formulario tipo `<input>` para introducir valores. Al pulsar el botón 'Añadir número' se incorporará el valor numérico al array correspondiente dependiendo si es par o impar.

Una vez almacenados los valores que el usuario ha decidido introducir se pulsará el botón 'Resolver' que mostrará por pantalla en dos cajas independientes los números pares e impares de cada uno de los arrays como se muestra en la imagen inferior.

- Será necesario utilizar dos arrays globales para almacenar los valores introducidos.
- Se dispondrá de un método ['isParImpar()'] que determine si el numero introducido es par o impar para añadirlo al array correspondiente.
- Los números capturados desde el elemento `<INPUT>` deberán ser convertidos a formato numérico antes de realizar el cálculo para almacenarlos en el array correspondiente.
- Se deberá realizar un control en la introducción de los valores para comprobar que realmente es un número. Si no fuese un valor numérico se advertirá mediante un mensaje informativo tipo alerta ['alert()'] al usuario y no se añadirá el valor a ninguno de los arrays.

*****Resultado*****

Escribir número:

Añadir

Resolver

NUMEROS IMPARES: 11, 27, 81, 15, 5, 33

NUMEROS PARES: 10, 24, 8, 16, 42, 30

Practica 29. (29_arrays_in_out.html) -> Método ['in_out_array()']

Disponemos de un array inicialmente vacío en el que iremos añadiendo o eliminando elementos por delante o por detrás del array. El ejercicio mostrará en todo momento el estado del array y cuál será su contenido cada vez que se realice una acción.

- Dispondremos de 4 botones y cada uno realizará una acción.
- Los métodos del objeto Array para utilizar serán '*shift*', '*unshift*', '*pop*', '*push*' y otros.
- El botón '*Reset*' dejará el array vacío y la caja sin contenido.
- Cuando se añada un valor se solicitará por teclado.
- Se mostrará siempre el ultimo valor que ha sido añadido y el último valor extraído independientemente si ha sido del inicio o del final del array.
- La zona eliminar permitirá escribir el nombre de un elemento del array y tras ser localizado lo eliminará del mismo. Si el valor no se encontrase dentro del array mostraría el mensaje de alerta advirtiéndolo por pantalla.
- Es posible utilizar más arrays si el alumno lo considera oportuno.

*****Resultado*****

Total Valores:	Juan, 11, true, 81, yellow, 5, 6.98	Añadir Delante
Añadir valor:	Input para introducir el valor	Añadir Detrás
Ultimo valor extraído:		Eliminar delante
Ultimo valor añadido:		Eliminar Detrás
		Reset
		Eliminar Valor: