

PRÁCTICA 3: INTERFACES GRAFICAS DE USUARIO – INTRODUCCIÓN AL JAVASCRIPT

Duración estimada: 100 min.

Objetivos:

- Iniciar al alumnado en el manejo básico del lenguaje javascript.
- Funciones y métodos
- Array
- Objetos predefinidos. Math, String, Object, etc

Recursos necesarios:

- Apuntes dados en clase.
- Guión de la práctica
- Internet.

Introducción:

1. **ARRAY:** Es un conjunto de datos ordenados por posiciones y todos asociados en una sola variable. Los datos pueden ser de cualquier tipo de dato, es decir, es posible crear una array que tenga una cadena en la primera posición, un número en el segundo, un objeto en el tercero y así sucesivamente. Podremos acceder a estos distintos datos de manera independiente o agrupándolos. Cabe resaltar que un array es un objeto, por ejemplo:

Formas de crear un Array

Existen dos formas de crear un Array y son las siguientes:

- La primera forma, nos permite crear un array colores estableciendo número de posiciones e indicando sus elementos como se muestra a continuación: **var colores = new Array();** De esta manera ya creamos el array colores, pero sin ninguna posición, entonces para establecer un número de posiciones para un array, sería: **var colores = new Array(10);**
- Veamos la segunda forma de crear un array, que es a través de los corchetes, aquí un ejemplo: **var colores = ["rojo", "azul", "verde"];** Acabamos de crear un array con tres datos string. En el caso de un array vacío sería: **var nombres = [];**

Métodos en los Arrays

- **Método length:** Si queremos conocer el número de elementos del array Podemos modificar el tamaño del array modificando esta propiedad. EJ: **colores.length = 2;**
- **Método toString():** Convierte todo el array en una cadena, separando cada elemento por comas, pero unidos en un string.
- **Métodos Stack:** Un array puede actuar como una pila, que te permite en un grupo de datos apilar y desapilar estos. A esta pila se conoce como el tipo LIFO(Last in, First out) último en entrar - primero en salir, lo que significa que el elemento más recientemente añadido es el primero en ser eliminado.



- **Método Push en Array.** Transforma un array añadiendo los elementos proporcionados y devolviendo la nueva longitud del array.
- **Método Pop en Array.** Este método permite eliminar el último elemento del array, disminuyendo la longitud de dicho array.
- **Métodos adicionales en los Arrays:**
 - **shift:** Permite obtener el primer elemento de un array
 - **unshift:** El método con el cual puedes agregar elementos al inicio del array, puede contener más de un argumento, esto quiere decir que puedes agregar en la misma sentencia más de un elemento. veamos su forma de uso.
 - **reverse:** método que establece que el array invierte sus elementos
 - **sort:** este método es muy útil cuando tengamos un array con elementos string, pues estos serán ordenados alfabéticamente. Lo ordenará alfabéticamente. De usarse este método en elementos de tipo numéricos nos devolverá datos incorrectos.
 - **slice:** método que puede contener uno o dos argumentos, que indiquen el inicio y parada de posiciones, pues devuelve los elementos contenidos en el array, de acuerdo a los argumentos indicados.
 - **splice:** cambia el contenido de un array removiendo elementos existentes y/o agregando nuevos elementos, hay tres maneras distintas de utilizar este método.
 - **supresión:** Cualquier número de elementos puede ser eliminado de la matriz especificando sólo dos argumentos: la posición del primer elemento que desea eliminar y el número de elementos a eliminar. Por ejemplo,


```
colores = ["amarillo", "anaranjado", "azul", "rojo", "verde"]
colores.splice(0, 2);
["azul", "rojo", "verde"]
```

Entonces, suprime los dos primeros elementos.
 - **inserción:** Se puede insertar elementos en el array en una posición específica, proporcionando tres argumentos: la posición donde deseas **agregar** elementos, el número de elementos que desea **eliminar** y el elemento a **insertar**. Opcionalmente, se puede especificar una cuarta, quinta, o cualquier número de otros parámetros a insertar. Por ejemplo,


```
var colores = ["amarillo", "anaranjado", "azul", "rojo", "verde"];
colores.splice(1,0,'pepe');
["amarillo", "pepe", "anaranjado", "azul", "rojo", "verde"]
```
 - **reemplazar:** Se comportaría como insertar en una posición específica al eliminar simultáneamente, para esto se tiene que especificar tres argumentos: la posición donde insertar, el número de elementos a borrar, y cualquier número de elementos a insertar. El número de elementos a insertar no tiene que coincidir necesariamente con el número de elementos que desea eliminar. Por ejemplo:


```
var colores = ["amarillo", "anaranjado", "azul", "rojo", "verde"];
colores.splice(2, 1, "morado", "violeta");
```

["amarillo", "anaranjado", "morado", "violeta", "rojo", "verde"]

Elimina un elemento en la posición 2 y luego inserta las cadenas "rojo" y "verde" en la matriz en la posición 2.

2. El objeto Math en JavaScript: El Objeto Math javascript nos permite trabajar con funciones matemáticas:

- $\text{Math.log}(x) = \ln(x)$
- $\text{Math.exp}(x) = e^x$
- $\text{Math.sqrt}(x)$ = raíz cuadrada de "x"
- $\text{Math.pow}(a, b) = ab$
- $\text{Math.floor}()$: número entero más cercano y menor
- $\text{Math.ceil}()$: número entero más cercano y mayor
- $\text{Math.round}()$: redondea al entero más próximo.
- $\text{Math.random}()$: número aleatorio entre 0 y 1
- $\text{Math.round}(y-x) * \text{Math.random}() + x$: número aleatorio entre "x" e "y".
- $\text{Math.sin}(x) = \sin(x)$ x en radianes
- $\text{Math.cos}(x) = \cos(x)$ x en radianes
- $\text{Math.tan}(x) = \text{tg}(x)$ x en radianes
- $\text{Math.atan}(x) = \text{arctg}(x)$ resultado en radianes
- $\text{Math.abs}(x)$: valor absoluto de "x"
- $\text{Math.max}(a,b)$: máximo valor de los dos
- $\text{Math.min}(a,b)$: mínimo valor de los dos



Secuencia y desarrollo: Realiza los siguientes scripts utilizando funciones.

1. **Ejercicio1** Realiza un programa que pida introducir un nombre de alumno por teclado, si el nombre del alumno existe en la lista implementada mediante un array, se visualizará el mensaje “Encontrado”, si no existe, se visualizará el mensaje de “No Encontrado”.
2. **Ejercicio2:** Escribir un programa que lea 10 números de teclado, los almacene en un array llamado numeros, y los escriba en orden inverso al que fueron introducidos. La entrada por teclado se hará con la orden **prompt()**.

Debes resolverlo mediante funciones. Para invocar la función **inverso**, debes utilizar un parámetros reset.

Por ejemplo, si escribimos los números 1,2,3,4,5,6,7,8,9 y 0 la salida que se generará será la siguiente:

NUMEROS INTRODUCIDOS INICIALMENTE:									
1	2	3	4	5	6	7	8	9	0

NUMEROS EN ORDEN INVERSO:									
0	9	8	7	6	5	4	3	2	1

3. **Ejercicio 3:** Realiza un programa que pide 6 números por pantalla y los almacena en un array. Debe realizar las siguientes acciones. Debes utilizar para cada acción una función.
 - Mostrar el array en el cuerpo de la página y en la consola.
 - Ordenarlos y mostrarlos
 - Invertir su orden y mostrarlos
 - Invertir su orden y mostrarlos
 - Mostrar el número de elementos que tiene el array
 - Búsqueda de un valor introducido por el usuario que nos diga si lo encuentra y su índice.