INSTITUTO UNIVERSITARIO AERONÁUTICO
Ingeniería Web II
Javascript - Arrays = Matrices
Docente: Ing. Carlos Simes

¿Qué es una matriz?

Una matriz es una variable especial, que puede contener más de un valor a la vez.

Si tienes una lista de elementos (una lista de nombres de automóviles, por ejemplo), el almacenamiento de los automóviles en variables individuales podría verse así:

```
var car1 = "Saab";
var car2 = "Volvo";
var car3 = "BMW";
```

Sin embargo, ¿qué pasa si quieres recorrer los autos y encontrar uno específico? ¿Y si no tuvieras 3 autos, sino 300? ¡La solución es una matriz!

Una matriz o array puede contener muchos valores con un solo nombre, y puede acceder a los valores haciendo referencia a un número de índice.

Crear una matriz

El uso de una matriz literal es la forma más fácil de crear una matriz de JavaScript.

```
var array_name = [item1, item2, ...];
```

Ejemplo

```
var cars = ["Saab", "Volvo", "BMW"];
```

Los espacios y los saltos de línea no son importantes. Una declaración puede abarcar varias líneas:

Ejemplo

```
var cars = [
  "Saab",
  "Volvo",
  "BMW"
];
```

Uso de la palabra clave JavaScript **new**

El siguiente ejemplo también crea una matriz y le asigna valores:

Ejemplo

```
var cars = new Array("Saab", "Volvo", "BMW");
```

Los dos ejemplos anteriores hacen exactamente lo mismo. No hay necesidad de usar new Array().

Por simplicidad, legibilidad y velocidad de ejecución, use el primero (el método literal de matriz).

Acceder a los elementos de una matriz

Accede a un elemento de matriz haciendo referencia al número de índice .

Esta declaración accede al valor del primer elemento en cars:

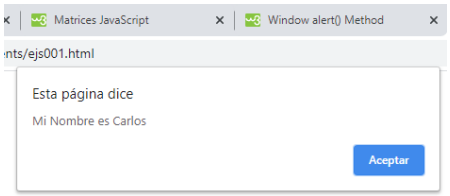
var name = cars[0];

Ejemplo

alert ('El valor del elemento 0 de la matriz es : ' + cars[0];)

Tips: Usamos la función alert("mensaje") de Javascript para mostrar mensajes al usuario e forma de cuadros de dialogo.

Por ejemplo: ***alert("Mi nombre es Carlos")*** *mostrará un cuadro de mensaje similar al de la imagen:*



Nota: Los índices de matriz comienzan con 0.

[0] es el primer elemento. [1] es el segundo elemento.

Cambiar un elemento de matriz

Esta declaración cambia el valor del primer elemento en cars:

cars[0] = "Opel";

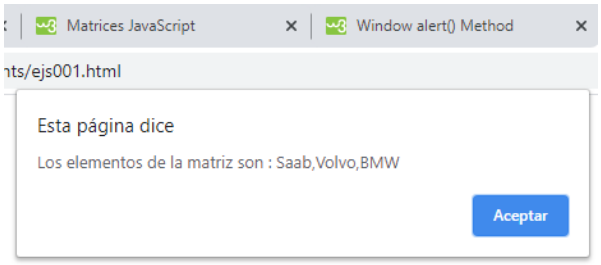
Ejemplo
var cars = ["Saab", "Volvo", "BMW"];
cars[0] = "Opel";

alert ('El valor del elemento 0 de la matriz es : ' + cars[0]);
Nos mostrará Opel.

Acceder a la matriz completa

Con JavaScript, se puede acceder a la matriz completa haciendo referencia al nombre de la matriz:

Ejemplo
var cars = ["Saab", "Volvo", "BMW"];
alert ('El valor del elemento 0 de la matriz es : ' + cars);
nos mostrará:



Las matrices usan números para acceder a sus "elementos". En este ejemplo, persona[0] devuelve Juan:

Matriz Persona

var persona = ["Juan", "Díaz", 46];
alert(persona[0]);

Los Objetos en javascript usan nombres para acceder a sus elementos.

En este ejemplo, persona.Nombre devuelve Juan:

Objeto Persona:

var persona = {Nombre:"Juan", Apellido:"Díaz", Edad:46};
alert(persona.Nombre);

Los elementos de matriz pueden ser objetos
Las variables de JavaScript pueden ser objetos. Las matrices son tipos especiales de objetos.

Debido a esto, puedes tener variables de diferentes tipos en la misma matriz.

Puedes tener objetos en una matriz. Puedes tener funciones en una matriz. Puedes tener matrices en una matriz:

myArray[0] = Date.now;
myArray[1] = myFunction;
myArray[2] = myCars;

Esta enorme flexibilidad del lenguaje de poder poner "cualquier cosa" dentro de una matriz es profundamente utilizada por los frameworks basados en javascript como **Angular, Vue.js, React**, etc y constituye una herramienta clave para la programación avanzada como veremos más adelante.

Propiedades y métodos de matriz

La verdadera fortaleza de las matrices de JavaScript son las propiedades y métodos integrados de la matriz:

Ejemplos

var x = cars.length; // Devuelve la longitud de la matriz
var y = cars.sort(); // Ordena el Array

La propiedad **length** de una matriz devuelve la longitud de una matriz (el número de elementos de la matriz).

Ejemplo

```
var frutas = ["Banana", "Naranja", "Manzana", "Mango"];
frutas.length;  // el largo de fruits es 4
```

Ojo !! La propiedad length siempre es uno más que el índice de matriz más alto, por lo tanto el último elemento de la matriz será

```
var ultimo = frutas[frutas.length - 1];
```

La forma más segura de recorrer una matriz es usando un lazo for:

Ejemplo

```
var frutas, texto, largo, i;
frutas = ["Banana", "Naranja", "Manzana", "Mango"];
largo = frutas.length;

texto = "";

for (i = 0; i < largo; i++) {
  texto += frutas[i]+ ", ";
}
texto += "";
```

También puedes usar la función **Array.forEach():**

Agregar elementos de matriz

La forma más fácil de agregar un nuevo elemento a una matriz es usar el método **push():**

Ejemplo

```
var frutas = ["Banana", "Naranja", "Manzana", "Mango"];
frutas.push("Limon");    // agrega un nuevo elemento a la matriz
```

El nuevo elemento también se puede agregar a una matriz utilizando la propiedad **length**:

Ejemplo
var frutas = ["Banana", "Naranja", "Manzana", "Mango"];
frutas[frutas.length] = "Limón"; // adds a new element (Lemon) to fruits

Atención ! Agregar elementos con índices altos puede crear "agujeros" indefinidos en una matriz

Ejemplo
var frutas = ["Banana", "Naranja", "Manzana", "Mango"];
frutas[6] = "Limón"; // agrega Limón como elemento 6 el 5 queda vacío.

Matrices asociativas

Muchos lenguajes de programación admiten matrices con índices con nombre.

Las matrices con índices nombrados se denominan **matrices asociativas** (o hashes).

JavaScript NO admite matrices con índices con nombre.

En JavaScript, las matrices **siempre usan índices numerados .**

Ejemplo
```
var person = [];
person[0] = "John";
person[1] = "Doe";
person[2] = 46;
var x = person.length;     // person.length will return 3
var y = person[0];         // person[0] will return "John"
```

ADVERTENCIA !!
Si usa índices con nombre, JavaScript redefinirá la matriz en un objeto estándar.
Después de eso, algunos métodos de matriz y propiedades producirán resultados incorrectos .

En JavaScript, **los objetos** usan índices con nombre .

Las matrices son un tipo especial de objetos, **con índices numerados.**

JavaScript no admite matrices asociativas.

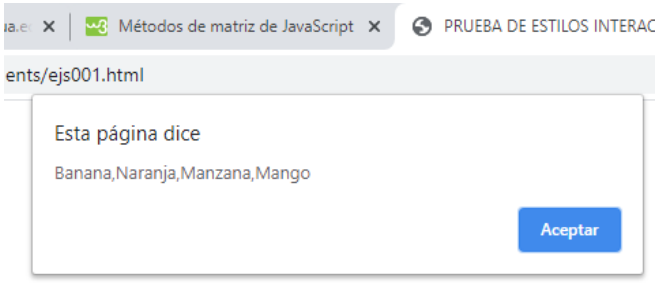
Métodos de matriz de JavaScript

Convertir matrices en cadenas

El método JavaScript **toString()**convierte una matriz en una cadena de valores de matriz (separados por comas).

Ejemplo
var frutas = ["Banana", "Naranja", "Manzana", "Mango"];
alert(frutas.toString());

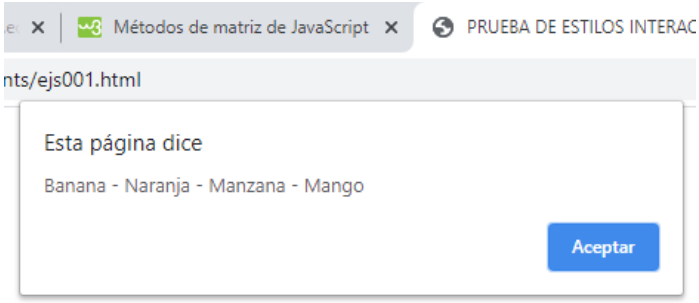
Nos mostrará:



El método **join()** también une todos los elementos de la matriz en una cadena.

Se comporta igual que **toString(),** pero además puede especificar el separador:

Ejemplo
var frutas = ["Banana", "Naranja", "Manzana", "Mango"];
alert(frutas.join(" - "));

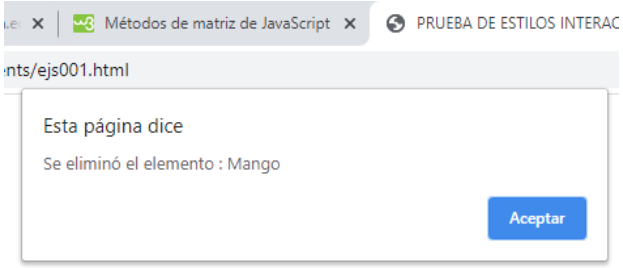


Cuando se trabaja con matrices, es fácil eliminar elementos y agregar nuevos elementos.

El método **pop()** elimina el último elemento de una matriz y devuelve dicho elemento:

Ejemplo

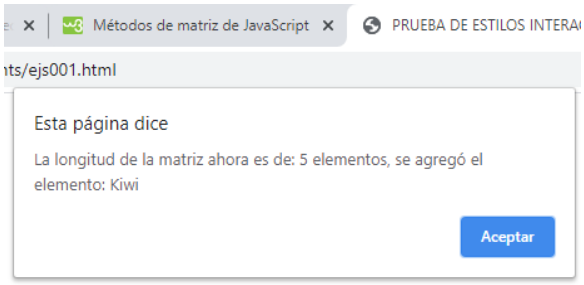
var frutas = ["Banana", "Naranja", "Manzana", "Mango"];
alert("se eliminó el elemento : " + frutas.pop()); // elimina Mango



El método **push()** agrega un nuevo elemento a una matriz (al final) y devuelve la nueva longitud de la matriz:

Ejemplo
var frutas = ["Banana", "Naranja", "Manzana", "Mango"];
alert("Se agregó el elemento : " + frutas[frutas.length-1] + ", la longitud de la matriz ahora es de: " + frutas.push("Kiwi") + " elementos");

Nos muestra:



El método **shift()** elimina el primer elemento de la matriz y "desplaza" todos los demás elementos a un índice inferior.

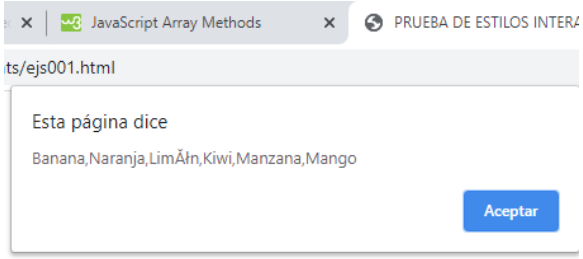
El método **unshift()** agrega un nuevo elemento a una matriz (al principio) y "desplaza" los elementos más antiguos hacia atrás.

Eliminar elementos
Dado que las matrices de JavaScript son objetos, los elementos se pueden eliminar mediante el operador **delete** de JavaScript:

Ejemplo
var frutas = ["Banana", "Naranja", "Manzana", "Mango"];
delete frutas[0]; // Atención: aquí se cambia el valor del elemento '0' por undefined creando un "agujero" en la posición '0'

Empalmar una matriz
El método splice() se puede usar para agregar nuevos elementos a una matriz:

Ejemplo
var frutas = ["Banana", "Naranja", "Manzana", "Mango"];
frutas.splice(2, 0, "Limón", "Kiwi");
alert(frutas);
nos mostrará:



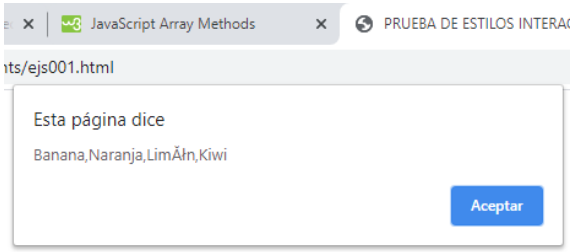
El primer parámetro (2) define la posición donde se deben agregar nuevos elementos (empalmados).

El segundo parámetro (0) define cuántos elementos deben eliminarse .

El resto de los parámetros ("Limón", "Kiwi") definen los nuevos elementos que se agregarán.

Con una configuración inteligente de parámetros, puede usar splice()para eliminar o reemplazar elementos sin dejar "agujeros" en la matriz:

var frutas = ["Banana", "Naranja", "Manzana", "Mango"];
frutas.splice(2, 2, "Limón", "Kiwi");
alert(frutas);
nos mostrará:



Donde se reemplazaron 2 elementos a partir de la posición 2.

El método **slice()** corta una parte de una matriz en una nueva matriz.

Este ejemplo corta una parte de una matriz a partir del elemento de matriz 1 ("Naranja"):

Ejemplo
var frutas = ["Banana", "Naranja", "Manzana", "Mango"];
var mini_frutas = frutas.slice(1);

El método **slice()** crea una nueva matriz. No elimina ningún elemento de la matriz fuente.

Este ejemplo corta una parte de una matriz a partir del elemento de matriz 1 ("Naranja"), por lo que mini_frutas contendrá **["Naranja", "Manzana", "Mango"];**

El método **slice()** puede tomar dos argumentos como slice(1, 3).

El método selecciona elementos del argumento inicial y hasta (pero sin incluir) el argumento final.

Ejemplo
var frutas = ["Banana", "Naranja", "Manzana", "Mango"];
var mini_frutas = frutas.slice(1,3);

Este ejemplo corta una parte de una matriz a partir del elemento de matriz 1 ("Naranja") y llegando hasta el elemento 3 sin incluirlo, por lo que mini_frutas contendrá **["Naranja", "Manzana"];**

Hasta aquí algunas de las funciones más empleadas de los array. Puede encontrar otras buscando en internet y el material provisto como material de referencia.

Actividades a desarrollar empleando matrices

Actividad Nº1:

Delta Software ha recibido un requerimiento para desarrollar una página web que permita cargar los datos de los alumnos de una escuela (Apellido, Nombre) y las calificaciones (3 por alumno).
En principio se podrán registrar hasta 5 alumnos a los que se les calculará el correspondiente promedio de las notas ingresadas.
Utilice HTML - CSS y Matrices para resolver lo siguiente:
Registrar los datos de los alumnos y sus calificaciones en una matriz.
Calcular los promedios de cada alumno, los que se guardarán también en la matriz.
Ordenar la matriz según los promedios de mayor a menor según los promedios desde el más bajo al más alto.
Mostrar los resultados en un tabla.

Actividad Nº2:

Agregue estilos a la actividad anterior para que una vez calculados los promedios, se muestren en verde los registros de los alumnos cuyo promedio sea >=7, en amarillo los promedios entre 4 y 6.99 y en rojo los promedios inferiores a 4.

Actividad Nº3:

Modifique la actividad anterior a fin de que en base a los promedios se registre en una columna adicional de la matriz, la condición de cada alumno la que será:
Promovidos => Promedio >=7
Regulares => Promedio entre 4 y 6.99
Libres => Promedio <4

Actividad Nº4:

Modifique la actividad anterior para que el usuario pueda al finalizar la carga de datos ingresar el modo de ordenamiento (promedio, apellido) y la tabla de resultados se muestre ordenada según el criterio ingresado.

Envíe las actividades al docente para su evaluación.