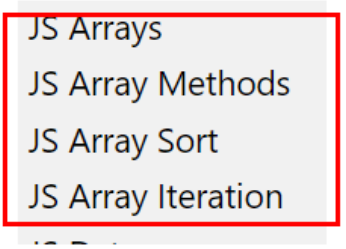
Realiza una **lectura comprensiva** de los apartados señalados en la imagen de w3Schools.



Algunas anotaciones a tener presentes:

- las matrices JavaScript siempre usan índices numéricos

- ***Ojo!! con las "matrices asociativas"; JavaScript NO admite matrices con índices de texto*** *(lo que en otros lenguajes se denomina "matrices asociativas"); si utilizas índices de texto, JavaScript va a redefinir la matriz como* ***un objeto*** *estándar. Y algunos de los métodos de matriz y propiedades producirán resultados incorrectos*:

  Ejemplo:

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

  … hasta aquí todo correcto, pero... si la definición la hacemos como en el ejemplo que sigue, el resultado **no es el esperado…!!**

```
var person = [];
person["firstName"] = "John";
person["lastName"] = "Doe";
person["age"] = 46;
var x = person.length;          // person.length devuelve 0
var y = person[0];              // person[0] devuelve undefined
```

- ***"¿Cuándo utilizar matrices y cuándo utilizar objetos?":***
  - Debes utilizar **objetos** cuando quieras que los índices de los elementos sean cadenas (de texto) .
  - Debes utilizar **matrices** cuando precises que los índices sean **números**.
  - Si no hay necesidad de usar el constructor new Array() utilizaremos [] en su lugar.

- ***Realiza la siguiente comprobación y obtén conclusiones:***

  Crea una matriz, msgMatriz, con una longitud de 0, luego asigna valores para msgArray[0] y msgArray[99], y comprueba cómo ha cambiado la longitud de la matriz unidimensional (de 1 a 100).

```
var msgMatriz = [];
```

```
console.log("la longitud inicial de la matríz es: ", msgMatriz.length);
msgMatriz[0] = "Hola";
msgMatriz[99] = "mundo";
console.log("la longitud final de la matríz es: ", msgMatriz.length);
```

❖ ***El método forEach() de Arrays***

Por último, presta especial atención al uso del método foreach como alternativa al siguiente código:

```
<script>
var fruits, text, fLen, i;
fruits = ["Banana", "Orange", "Apple", "Mango"];
fLen = fruits.length;

text = "<ul>";
for (i = 0; i < fLen; i++) {
  text += "<li>" + fruits[i] + "</li>";
}
text += "</ul>";

document.getElementById("demo").innerHTML = text;
</script>
```

para obtener:

- Banana
- Orange
- Apple
- Mango

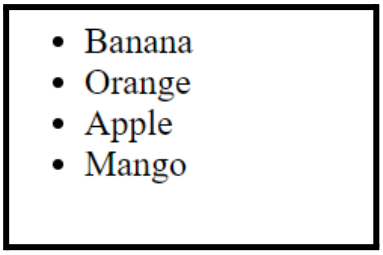
Podemos hacer el mismo bucle utilizando el método ***forEach()*** del array, al cuál le pasamos una *función callback*.

```
<script>
var fruits, text;
fruits = ["Banana", "Orange", "Apple", "Mango"];

text = "<ul>";
fruits.forEach(myFunction);
text += "</ul>";
document.getElementById("demo").innerHTML = text;

function myFunction(value) {
  text += "<li>" + value + "</li>";
}
</script>
```

que también obtiene:



Funciones callback no son más que un tipo de funciones que se pasan por parámetro a otras funciones. La función callback que le hemos pasando a forEach() se va a ejecutar por cada uno de los elementos del array y en cada iteración el parámetro value va a tener un valor diferente (el elemento del array)

Revisa los ejemplos de este enlace para más detalle sobre dicho método

Tarea:

*En los casos en los que el enunciado del problema sea ambiguo o no sea lo suficientemente completo, el alumno debe recoger **mediante comentarios JavaScript** las suposiciones que se adopten indicando las razones de su elección y/o decisiones.*

1. Crea un script donde declares un **array** vacío denominado ***nombres***. Pide al usuario tres nombres usando la sentencia ***prompt*** y almacena esos nombres como elementos 0, 1 y 2 del array. A continuación muestra en pantalla el contenido del array.

2. Genera un script que pida cinco números que guardarás en un **array.** Recorre el array para mostrar el resultado de multiplicar cada uno de los números por 5. *Ejemplo: Se pedirán al usuario cinco números, supongamos que introduce 1, 3, 9, 10 y 7. A continuación se mostrará el mensaje:*

   *Multiplicamos por 5 los números introducidos: 1*5 = 5, 3*5 = 15, 9*5 = 45, 10*5 = 50 y 7*5 =35.*

3. Leer una secuencia de n números. Almacenarlos en un **vector** y mostrar la posición donde se encuentra el mayor valor leído.

4. Dados dos **vectores** A y B de n elementos cada uno, obtener un vector C donde la posición i almacene la suma de A[i]+B[i].

5. Dado un **vector** de secuencias de caracteres mostrar la longitud de cada una de ellas.

6. Definir un **array** y pasarlo como parámetro a una función *Cargar* que servirá para cargarlo con los valores correspondientes a los sueldos de 5 empleados. Una segunda función *calcularGastos()* recibirá como parámetro el array y obtendrá la suma de todos los sueldos. La página debe listar todos los sueldos y el total de sueldos.

7. Por último, realiza la batería de pruebas que estimes oportuna para asegurarte que sabes utilizar los ***métodos de array*** propuestos en el tutorial