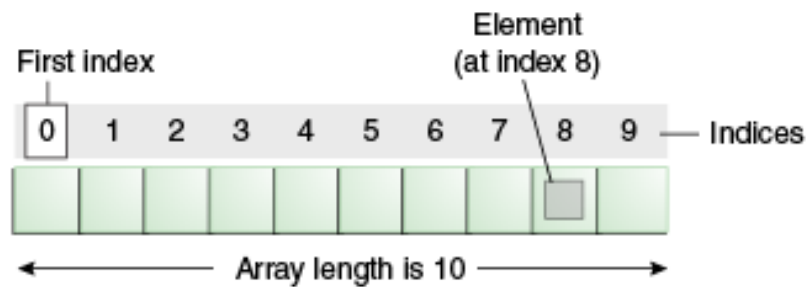


## MÓDULO 2: ARREGLOS Y MATRICES EN JAVASCRIPT



- Un vector, arreglo o colección es una secuencia de posiciones de la memoria central del PC (RAM) que contiene datos y se puede seleccionar individualmente mediante el uso de índices.
- Una analogía con el mundo real son los estantes, las filas de espera, etc.
- Tienen una sola dimensión y múltiples posiciones.
- Cada posición puede contener diversos tipos de datos.
- **La posición inicial siempre será 0** (por defecto, pero se puede cambiar por parte del programador)



## DEFINICIÓN DE ARREGLOS

```
1 //USANDO EL METODO CONSTRUCTOR DE LA API DE JS
2 let Heroes = new Array("Batman", "Superman", "IronMan", "Ultraman", "Deadpool");
3
4 //DEFINIENDO ARREGLOS DE FORMA LITERAL
5 let Frutas = ["uva", "mango", "manzana", "papaya", "sandia", "kiwi"];
6
```

Un arreglo “mezclado”:

```
const arregloMix = [74, "colombia", false, "cali", 1200];
console.log(arregloMix);
```

## CREANDO ARREGLOS CON “HUECOS”

```
1 let sillas = [, , , , ,];
2 console.log(sillas);
3
```

## MANIPULACIÓN DE ARREGLOS

Llenado manual: llamando al índice de la posición

```
const ciudades = [];

ciudades[3] = "New york";
ciudades[2] = "Cali";
ciudades[0] = "Pereira";
ciudades[1] = "Manizales";
console.log(ciudades);
```

- Borrando el ultimo: ciudades.pop()
- Borrando el primero: ciudades.shift()
- Agregando al final: ciudades.push("yumbo", "buga")
- Agregando al principio: ciudades.unshift("tulua", "guacarí", "vijos")

ANALISIS DEL SPLICE:

## miArreglo.splice(**p**,**p**,"dato","dato".....)

**p**: posición del arreglo para hacer la operación de inserción o borrado

**p**: numero de elementos a remover

- ejemplo insertando: Insertando en cualquier sitio del arreglo: `ciudades.splice(3,0,"Cartago")`
- ejemplo borrando: Insertando en cualquier sitio del arreglo: `ciudades.splice(0,3)`
- se puede combinar borrado e inserción: `ciudades.splice(0,3,"lima","quito")`

## RECORRIENDO O ITERANDO LOS ARREGLOS

Propiedad importante: **.length** = retorna la longitud del arreglo

### FORMA CLÁSICA: USANDO EL BUCLE FOR

```
const ciudades = [];  
  
ciudades[3] = "New york";  
ciudades[2] = "Cali";  
ciudades[0] = "Pereira";  
ciudades[1] = "Manizales";  
  
for (let index = 0; index < ciudades.length; index++) {  
  console.log(`Codigo: ${index} nombre: ${ciudades[index]}`);  
}
```

### USANDO FOREACH

```
ciudades.forEach((ciudad) => {  
  console.log(ciudad);  
});
```

Foreach utiliza una función callback (una función llamada dentro de otra) que recibe un parámetro que corresponde a la posición de cada elemento



## USANDO EL METODO MAP (IMPORTANTE)

```
let impares = [1, 3, 5, 7, 9];  
  
let pares = impares.map((impar) => {  
  return impar * 2;  
});  
console.log(pares);
```

el método map crea un nuevo array usando una función callback con el resultado de una operación (si se requiere)