

# Bootcamp Java Developer

Fase 2 - Java Web Developer  
Módulo 18



# Arreglos y objetos

# Arreglos y objetos

Al comprender cómo se trabaja con arreglos y objetos, se pueden integrar ambos conceptos, como se ve en el ejemplo.

Tenemos un listado de empleados. Estos empleados tienen **diferentes propiedades**, pero a su vez **son varios y queremos tenerlos en una misma variable (arreglo)**, lo que haremos será:

```
<script>
var empleados = [{nombre: 'Juan', apellido : 'Pedro',
edad: 40}, {nombre: 'Ana', apellido : 'García',
edad: 25}, {nombre: 'Maria', apellido : 'Zarate',
edad: 34}]

</script>
```

# Métodos útiles para trabajar con arrays

Existen algunos **métodos útiles** para trabajar con arreglos.

Por ejemplo, de esta manera se puede **ordenar nuestro arreglo**:

```
empleados.sort()
```

También se puede **acceder rápidamente al último dato**:

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



Si quisiéramos **mostrar los pasos anteriores en una alerta**, haremos esto:

```
1 var ultimo = empleados[empleados.length - 1];  
2 console.log(ultimo);
```

De esta manera, es posible hacer lo siguiente:

```
1 var empleados = [  
2   { nombre: 'Juan', apellido: 'Pedro', edad: 40 },  
3   { nombre: 'Ana', apellido: 'Garcia', edad: 25 },  
4   { nombre: 'María', apellido: 'Zarate', edad: 34 },  
5 ];  
6  
7 var ultimo = empleados[empleados.length - 1];  
8 console.log(ultimo);  
9
```

Si quisiéramos **mostrar los pasos anteriores en una alerta:**

```
var ultimo = empleados[empleados.length - 1];  
alert(ultimo)
```

De esta manera, es posible hacer lo siguiente:

```
<script>  
var empleados = [{nombre: 'Juan', apellido : 'Pedro',  
  edad: 40}, {nombre: 'Ana', apellido : 'García',  
  edad: 25}, {nombre: 'Maria', apellido : 'Zarate',  
  edad: 34}]  
  
var ultimo = empleados[empleados.length - 1];  
alert(ultimo.nombre)  
  
</script>
```



Mediante el método **push** se puede **agregar un dato al final del arreglo**.

Por ejemplo, generamos:

```
var empleados = ['Juan', 'Roberto']
```

Para **agregar un dato al final**, lo haremos así:

```
<script>  
var empleados = ['Juan', 'Roberto']  
  
empleados.push('Maria')  
  
</script>
```

Luego se muestra por **consola**:

```
1 console.log(empleados);  
2
```

El método **pop** permite **quitar un dato**. Por ejemplo:

```
1  var empleados = ['Juan', 'Roberto'];  
2  
3  empleados.pop();  
4  
5  console.log(empleados);  
6
```

El resultado será el siguiente:

Juan

También podemos **sumar un dato al principio del arreglo** lo haremos con **unshift**:

```
1  var empleados = ['Juan', 'Roberto'];  
2  
3  empleados.unshift('Maria');  
4  
5  console.log(empleados);  
6
```

El resultado en la consola será el siguiente:

Maria,Juan,Roberto



Por otro lado, si **quiero quitar un dato del inicio**, lo haré de la siguiente forma, con **shift**.

```
1  var empleados = ['Maria', 'Juan', 'Roberto'];  
2  
3  empleados.shift();  
4  
5  console.log(empleados);  
6
```



## Método Splice

Si no queremos **quitar datos** exactamente al inicio o al final, lo que podemos hacer es lo siguiente:

```
1  var empleados = ['Juan', 'Roberto', 'Maria', 'Ana'];  
2  
3  empleados.splice(2);  
4  
5  console.log(empleados);  
6
```

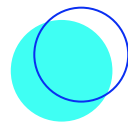
El bloque anterior removerá el elemento cuyo índice es 2 (dos) en adelante.

Para indicar **cuántos elementos serán removidos** con el método **splice** se agrega:

```
1  var empleados = ['Juan', 'Roberto', 'Maria', 'Ana'];  
2  
3  empleados.splice(2, 1);  
4  
5  console.log(empleados);  
6
```

El resultado será:

Juan,Roberto,Ana

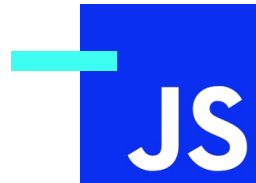


No solo podemos hacer esto, también podemos **quitar datos y, al mismo tiempo, agregar otros en esa misma posición:**

```
1  var empleados = ['Juan', 'Roberto', 'Maria', 'Ana'];  
2  
3  empleados.splice(2, 1, 'Carla');  
4  
5  console.log(empleados);  
6
```

El resultado será:

Juan,Roberto,Carla,Ana



## Método Concat

Se pueden **unir diferentes arreglos**. Por ejemplo, suponemos que hay dos sectores:

```
<script>
var sectorA = ['Juan', 'Roberto', 'Maria', 'Ana']
var sectorB = ['Pedro', 'Carlos', 'Rosa', 'Carla']

</script>
```

Pero por alguna razón **queremos unir ambos arreglos**:

```
<script>
var sectorA = ['Juan', 'Roberto', 'Maria', 'Ana']
var sectorB = ['Pedro', 'Carlos', 'Rosa', 'Carla']
var sectores = sectorA.concat(sectorB);

</script>
```

Para **mostrarlo en un contenedor**:

```
1  var sectorA = ['Juan', 'Roberto', 'Mario', 'Ana']
2  var sectorB = ['Pedro', 'Carlos', 'Rosa', 'Carla']
3  var sectores = sectorA.concat(sectorB);
4
5  console.log(sectores);
6
```

El resultado será:

Juan,Roberto,Maria,Ana,Pedro,Carlos,Rosa,Carla



## Método Slice

Permite **partir un arreglo a partir de cierta posición**, por ejemplo:

```
1 var sectorA = ['Juan', 'Roberto', 'Mario', 'Ana']
2
3 var mujeres = sectorA.slice(2);
4
5 console.log(mujeres);
6
```

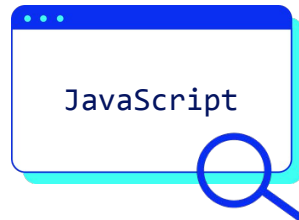
De esta manera indicamos que queremos a partir de la posición 2. El resultado será:

Maria,Ana



# Revisión

- Repasar el concepto de **arreglo más objeto**.
- Mostrar una **lista de empleados y diferentes propiedades**.
- Estudiar los métodos **mostrados en el contenido**.
- Practicar los **métodos** para poder entenderlos de mejor forma.
- Trabajar en el **Proyecto Integrador**.
- Realizar las preguntas necesarias al/la docente antes de continuar.





**¡Sigamos  
trabajando!**