

Introducción a la programación en JavaScript

Módulo 3 - Arreglos & Objetos

Arreglos & Objetos

Arreglos & Objetos

Entendiendo el trabajo tanto con arreglos como con objetos, podemos integrar ambos conceptos de la siguiente manera.

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 lo siguiente:

```
<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

Otra cosa que también podemos hacer, es utilizar algunos **métodos útiles** para trabajar con arreglos. Por ejemplo:

```
empleados.sort()
```

De esta manera podemos ordenar nuestro arreglo, también podemos acceder rápidamente al último dato del arreglo, por ejemplo:

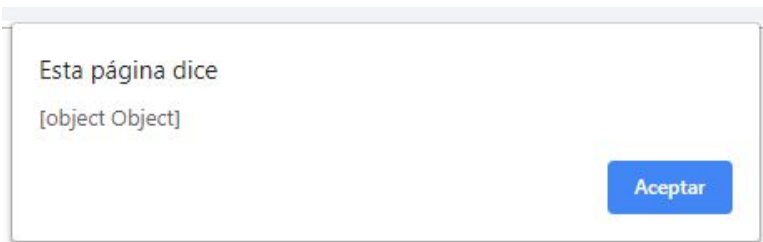
```
var ultimo = empleados[empleados.length - 1];
```

Métodos útiles para trabajar con arrays

Si los pasos anteriores queremos mostrarlos en una alerta, haremos lo siguiente:

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

Por supuesto nos encontraremos nuevamente con el problema planteado anteriormente.



Métodos útiles para trabajar con arrays

De esta manera podemos hacer lo siguiente:

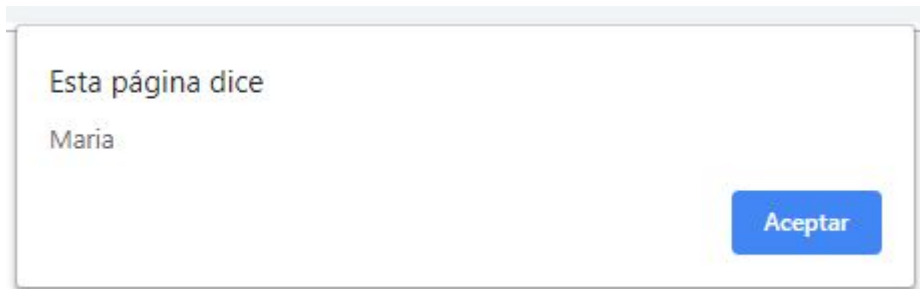
```
<script>
var empleados = [{nombre: 'Juan', apellido : 'Pedro',
  edad: 40}, {nombre: 'Ana', apellido : 'García',
  edad: 25}, {nombre: 'Maria', apellido : 'Zarate',
  edad: 34}]

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

</script>
```

Métodos útiles para trabajar con arrays

Lo anterior se mostrará en pantalla de esta manera:



Esta página dice

Maria

Aceptar

También podemos trabajar con un método llamado *push*, de esta forma podemos agregar un dato al final del arreglo, por ej., generamos el siguiente arreglo:

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

Métodos útiles para trabajar con arrays

Para **agregar un dato al final**, lo haremos de la siguiente forma:

```
<script>
var empleados = ['Juan', 'Roberto']

empleados.push('Maria')

</script>
```

Luego lo mostraremos **en una caja**:

```
<div id="caja"> </div>

<script>
var empleados = ['Juan', 'Roberto']

empleados.push('Maria')

document.getElementById('caja').innerHTML = empleados

</script>
```


Métodos útiles para trabajar con arrays

El resultado en pantalla desde nuestro **navegador** será el siguiente:

Juan,Roberto,Maria

También podemos trabajar con el método **pop**.
Este método nos permite sacar un dato,

```
<script>
var empleados = ['Juan', 'Roberto']

empleados.pop()

document.getElementById('caja').innerHTML = empleados

</script>
```

Métodos útiles para trabajar con arrays

Si con el método **pop** sacamos un dato el resultado será el siguiente:

Juan

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

```
<div id="caja"> </div>

<script>
var empleados = ['Juan', 'Roberto']

empleados.unshift('Maria')

document.getElementById('caja').innerHTML = empleados

</script>
```

Métodos útiles para trabajar con arrays

El resultado en el navegador será el siguiente:

María,Juan,Roberto

En este caso **María se agrega al principio, no al final**. Por su lado, si quiero quitar a María (quitar un dato del inicio, lo haré de la siguiente forma, con **shift**.

```
<div id="caja"> </div>

<script>
var empleados = ['Juan', 'Roberto']

empleados.shift()
document.getElementById('caja').innerHTML = empleados

</script>
```

Método Splice

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

```
<div id="caja"> </div>

<script>
var empleados = ['Juan', 'Roberto', 'Maria', 'Ana']

empleados.splice(2)
document.getElementById('caja').innerHTML = empleados

</script>
```

Método Splice

Removerá en este caso el elemento **cuyo índice es 2(dos) en adelante.**

Si además queremos indicar cuántos elementos serán removidos, **agregaremos algo más:**

```
<body>
  <div id="caja"> </div>
  <script>
var empleados = ['Juan', 'Roberto', 'Maria', 'Ana']

empleados.splice(2,1)
document.getElementById('caja').innerHTML = empleados

  </script>
</body>
```

Método Splice

En base al código anterior el resultado será:

Juan,Roberto,Ana

No sólo podemos hacer esto, sino que también podemos al mismo tiempo qué quitamos, agregar datos en esa misma posición:

```
<div id="caja"> </div>

<script>
var empleados = ['Juan', 'Roberto', 'Maria', 'Ana']

empleados.splice(2,1, 'Carla')
document.getElementById('caja').innerHTML = empleados

</script>
```

Método Splice

En base al código anterior, el resultado será: Juan,Roberto,Carla,Ana

Esto anterior es:

```
<div id="caja"> </div>

<script>
var empleados = ['Juan', 'Roberto', 'Maria', 'Ana']

empleados.splice(2,1, 'Carla')
//(posicion para remover, cuantos a partir de esa
//posición, datos que agrego )
document.getElementById( 'caja' ).innerHTML = empleados

</script>
```

Método Concat

También se pueden **unir diferentes arreglos**, por ejemplo, vamos a suponer que tenemos dos sectores:

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

</script>
```

Pero por alguna razón, sólo en algún momento preciso **queremos unir ambos arreglos**, lo haremos de la siguiente forma:

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

</script>
```


Método Concat

Si lo queremos mostrar en un contenedor lo haremos de la siguiente forma:

```
<div id="caja"> </div>

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

document.getElementById('caja').innerHTML = sectores
</script>
```

El resultado será el siguiente:

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

Método Slice

Slice nos **permite partir un arreglo a partir de cierta posición**, por ejemplo:

```
<div id="caja"> </div>

<script>
var sectorA = ['Juan', 'Roberto', 'Maria', 'Ana']

var chicas = sectorA.slice(2)

document.getElementById('caja').innerHTML = chicas;
</script>
```

De esta manera indicamos qué queremos partir el arreglo a partir de la **posición 2**, de esta forma el resultado será el siguiente:

Maria,Ana

Revisión

- Repase el concepto de **arreglo más objeto**.
- Trabaje mostrando una **lista de empleados y diferentes propiedades**.
- Estudie los métodos **mostrados en el contenido**.
- Practique los **métodos** para poder entenderlos de mejor forma.
- Trabaje en el **Proyecto Integrador**.
- Realice las preguntas necesarias al/la docente antes de continuar.



¡Muchas gracias!

¡Sigamos trabajando!