

Introducción a la programación en JavaScript

Módulo 3 - Arreglos

Arreglos

Arreglos

Los arreglos no son más ni menos que un **tipo especial de variable** que puede contener más de un valor al mismo tiempo.

Los arreglos pueden contener **los mismos valores** que vimos anteriormente.

Es decir, pueden contener **variables numéricas, string o booleanas**. También es importante entender que no necesariamente deben ser todas del mismo tipo. Por ejemplo, un arreglo puede contener una variable **tipo string y otra numérica**.

```
<script>  
  
var arreglo = ['juan', 'marcelo', 20, 40]  
  
</script>
```

Diferentes formas de trabajar con arreglos

Otra forma de trabajar con arreglos es la que se verá en el ejemplo a la derecha. Sin embargo, la primera forma es más sencilla.

Veamos aquí un ejemplo de la forma que se solía trabajar anteriormente a través de **new Array**.

```
<script>  
  
var arreglo = new Array('juan', 'marcelo', 20, 40);  
  
</script>
```

Mostrar los datos de un arreglo

Por supuesto, es importante mostrar los **datos contenidos en un array**. Lo haremos de la siguiente manera:

Es decir que siempre **identificamos al dato**, por encontrarse en tal o cual posición del arreglo. Por ejemplo, siempre el índice del **primer elemento es 0**.

```
<script>  
  
var arreglo=['juan', 'marcelo', 20, 40];  
alert(arreglo[0]);  
  
</script>
```

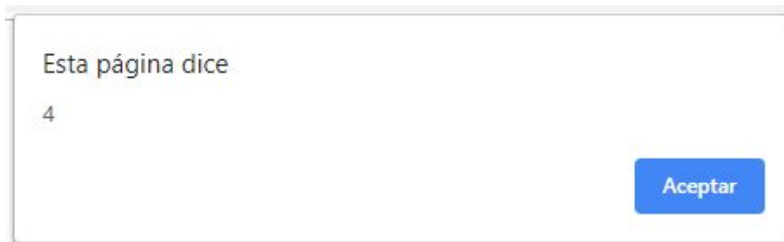
```
<script>  
  
var arreglo=['juan', 'marcelo', 20, 40];  
alert(arreglo[3]);  
  
</script>
```

Mostrar los datos de un arreglo

Claro está que también puedo obtener la **cantidad de datos que hay un arreglo**:

```
<script>  
  
var arreglo = ['juan', 'marcelo', 20, 40];  
alert(arreglo.length);  
  
</script>
```

El resultado será el siguiente:



Mostrar los datos de un arreglo

Si quisiera mostrar absolutamente todos los **datos del arreglo al mismo tiempo**, lo haría de la siguiente forma:

```
<body>

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

  <script>

var arreglo =['juan', 'marcelo', 20, 40];
document.getElementById("caja").innerHTML = arreglo;

  </script>

</body>
```

Mostrar los datos de un arreglo

Al mostrar todos los datos del arreglo, el resultado en pantalla será el siguiente:

juan,marcelo,20,40

Vamos a suponer que estamos trabajando con un arreglo donde mencionamos los nombres de los empleados de una empresa:

```
<script>
var arreglo = ['juan', 'marcelo', 'ana', 'luis'];
</script>
```


Mostrar los datos de un arreglo

Si los pasos anteriores se quieren mostrar en pantalla haremos lo siguiente:

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

<script>

var arreglo =['juan', 'marcelo', 'ana', 'luis'];
document.getElementById('caja').innerHTML = arreglo

</script>
```

El resultado será: **juan,marcelo,ana,luis**

Objetos

Objetos y su definición

Los objetos pueden **contener propiedad y valores**, por lo tanto nos son más sencillos para poder indagar características de un elemento más complejo que las variables consideradas anteriormente.

Por ejemplo:

```
<script>  
  
var alumno = { nombre: 'Ivan', apellido: 'García'}  
  
</script>
```



Objetos y su definición

Si queremos mostrar la información del objeto anterior llamado *alumno*, haremos lo siguiente:

```
<script>

var alumno = { nombre: 'Ivan', apellido: 'García'}
alert(alumno)

</script>
```

El resultado será el siguiente:

Esta página dice
[object Object]

Aceptar

Objetos y su definición

El problema fundamental, es que estamos **trayendo al objeto entero y necesitamos en todo caso mostrar partes del mismo**, ahora el resultado en pantalla será:

Esta página dice

Ivan

Aceptar

Esta página dice

García

Aceptar

Revisión

- Repase el concepto de **arreglo en Javascript**.
- Trabaje mostrando una **lista de empleados**.
- Marque la diferencia entre objetos con **propiedades y arreglos**.
- Avance con el output a través de **innerHTML()**
- Implemente **document.getElementById()** para acceder al **DOM**.
- Trabaje en el **Proyecto Integrador**.
- Realice las preguntas necesarias al/la docente antes de continuar.



¡Muchas gracias!

¡Sigamos trabajando!