

Introducción a la programación en JavaScript

Módulo 3 - Bucles

Bucles

For

Vamos a suponer que queremos recorrer un arreglo de empleados, por ejemplo, el siguiente:

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

</script>
```

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

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

mensaje += empleados[0] + "<br>";
mensaje += empleados[1] + "<br>";
mensaje += empleados[2] + "<br>";
mensaje += empleados[3] + "<br>";
mensaje += empleados[4] + "<br>";
mensaje += empleados[5] + "<br>";

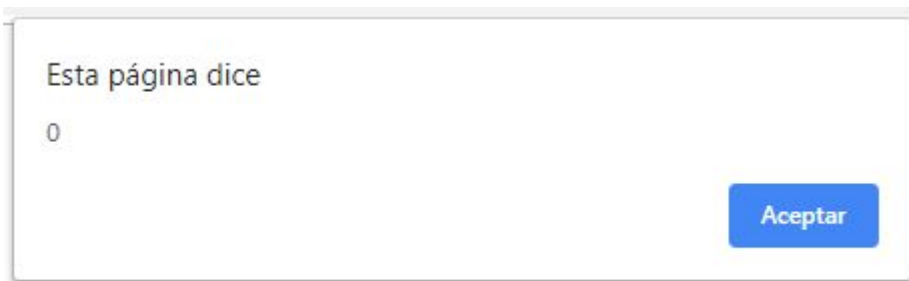
document.getElementById(caja) = mensaje;
</script>
```

For

La forma anterior es tediosa y molesta para trabajar, de la siguiente manera será mucho más sencilla.

```
for(var i = 0; i < 2; i++)  
  alert(i)
```

El resultado será el siguiente:



For

Y parar en ese dato, pues hemos fijado que i nunca **debe ser mayor que 2**. De esta manera, lo que hacemos es repetir un ciclo (aumentar en este caso i) en tanto y en cuanto el **mismo no supere a 2**.

La realidad es que este ejemplo anterior **carece de sentido**, a menos que lo utilicemos, por ejemplo, para cuestiones más prácticas como **iterar un arreglo**.

```
<script>
var empleados = ['Juan', 'Roberto', 'Maria', 'Ana']
var mensaje = "";

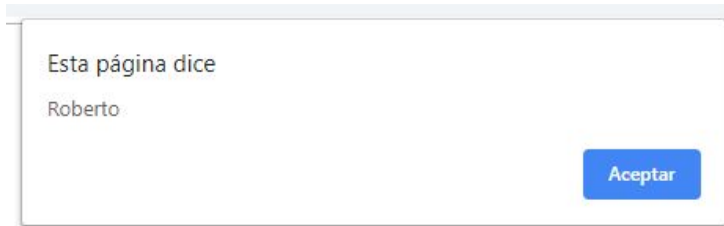
for(var i = 0; i < 2; i++)

  alert(empleados[i])

</script>
```

For

El resultado será perfecto, y nos mostrará el nombre de cada empleado siempre que no supere al índice **del arreglo 2**, es decir nos **mostrará hasta Roberto**:



Para mostrarlo en un contenedor:

```
<script>
var empleados = ['Juan', 'Roberto', 'Maria', 'Ana']
var mensaje = "";

for(var i = 0; i < 2; i++)
  mensaje = empleados[i]
document.getElementById('caja').innerHTML = mensaje;

</script>
```

For

Claramente no es lo que esperamos, **porque en realidad, el problema reside en que a diferencia de una alerta**, todo el tiempo reemplazamos el contenido de la caja, y nos muestra el último valor, lo que debemos hacer es lo siguiente, utilizar un operador de asignación que ya hemos visto:

```
<body>

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

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

for(var i = 0; i < 2; i++)
  mensaje += empleados[i]
document.getElementById('caja').innerHTML = mensaje;

  </script>

</body>
```

For

El resultado del código anterior será el siguiente:

JuanRoberto

Claro que no queda del todo bien en la presentación, por lo tanto, lo que haremos será agregar algunos elementos **de HTML, por ejemplo:**

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

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

for(var i = 0; i < 2; i++)
mensaje += empleados[i] + '<br>'
document.getElementById('caja').innerHTML = mensaje ;

</script>
```


For

Por otro lado, ¿qué pasaría si, en realidad, esto nos limita demasiado y quizás ese arreglo aumenta de manera dinámica? ¿cómo haremos?

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

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

for(var i = 0; i < empleados.length ; i++)

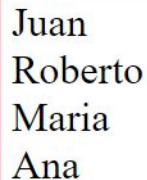
mensaje += empleados[i] + '<br>'

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

</script>
```

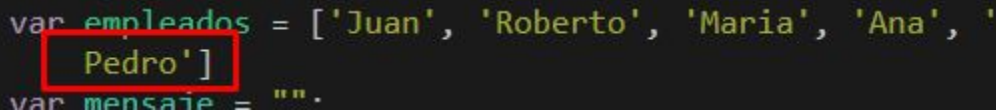
For

De esta manera siempre , se adaptará la condición a la cantidad de datos que tenga en el arreglo, por ejemplo:



Juan
Roberto
Maria
Ana

Luego puedo hacer lo siguiente, es decir **agrego a Pedro**, el contenedor automáticamente mostrará todos los datos del arreglo, sin necesidad que se deba variar el límite:



```
var empleados = ['Juan', 'Roberto', 'Maria', 'Ana', 'Pedro'];  
var mensaje = ''.
```

While

While

El bucle **while** es similar, trabajando su **sintaxis de la siguiente manera:**

```
while (i < 10) {  
  mensaje += "El dato es " + i;  
  i++;  
}
```

Este bucle si bien parecido al **for**, tiene ciertas diferencias que son importantes establecer.



While

Por ejemplo, su sintaxis será la siguiente:

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

<script>
var empleados = ['Juan', 'Roberto', 'Maria', 'Ana', '
    Pedro']
var mensaje = "";
var i = 0;
while(i < empleados.length) {

    mensaje += empleados[i] + '<br>'

    i++
}

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

</script>
```

Iteración de arreglos con objetos

Si por ejemplo nuestro arreglo tiene objetos y queremos recorrerlo como hemos aprendido lo haremos de la siguiente forma:

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

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

var mensaje = "";
var i = 0;
while(i < empleados.length) {

mensaje += empleados[i].nombre + '<br>'

i++
}

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

</script>
```

Revisión

- Repase el concepto de **bucle**.
- Trabaje mostrando datos de un arreglo con **for**.
- Trabaje mostrando datos de un arreglo con **while**.
- Muestre datos de un **arreglo con objetos** que **contengan propiedades**.
- Trabaje en el **Proyecto Integrador**.
- Realice las preguntas necesarias al/la docente antes de continuar.



¡Muchas gracias!

¡Sigamos trabajando!