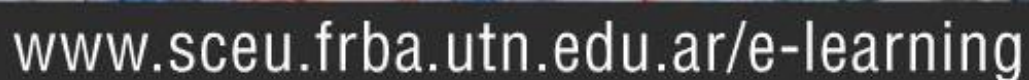




Desarrollo Web

con JavaScript





Temario:

Temario:

1.Vectores o Arrays

2. Metodos

3.Bibliografía

1. Vectores o Arrays

En la unidad anterior hablamos sobre las funciones que podemos realizar utilizando valores pero solamente lo hicimos manipulando valores únicos. Sin embargo, no siempre va a ser el caso.

En varias situaciones nos vamos a encontrar con una gran cantidad de valores que manipular y hacerlo de forma individual hace que sea un trabajo engorroso y bastante lento. Por esta situación, JavaScript nos ofrece la posibilidad de agrupar esta información para que su manipulación sea más práctica. Esto lo hace utilizando lo que llamamos vectores o más comúnmente conocidos como Arrays.

Los Arrays son estructuras que nos permiten almacenar una gran cantidad de información en forma de una lista, donde mediante distintos métodos vamos a poder tratar a cada uno de los valores que lo componen.

Algunas de las cualidades más importantes de estas estructuras son:

- La posibilidad de almacenar cualquier tipo de dato: variables, valores e incluso otros arrays, funciones u otro tipo de estructuras.
- La posibilidad de tener en un mismo array una combinación de varios tipos de datos y valores.
- No poseer un límite en cuanto a la cantidad de datos que puede tener, e incluso la posibilidad de ir modificando su longitud, a lo largo del código de JavaScript.
- Los arrays se componen por un par de corchetes (`[]`) que engloban a la lista, la que se escribe como una estructura de cubículos donde colocamos a cada uno de sus valores uno al lado del otro, separados por una coma (`,`).
- Para poder utilizarlos, guardamos estos arrays en una variable.

A continuación, veremos un ejemplo de la sintaxis de un array.

SINTAXIS

```
var lista =[ "Juan",11224, funcion(), unaVariable, true]
```

Para poder identificar a cada uno de los datos que compone al array, contamos con algo que se llama **índice** que asigna a cada dato o valor un número. Esta numeración empieza desde el 0 y le da un posicionamiento a cada valor; es decir que el primer valor del array contiene la posición o índice 0, el siguiente la posición o índice 1 y así sucesivamente.

Tenemos que tener en cuenta que este índice se asigna automáticamente, es decir que no tenemos que programarlo nosotros. En caso que cambiemos de lugar los valores, el orden del índice permanece, por lo que el valor ahora tendrá un nuevo índice.

EJEMPLO

```
/*   indice:    0      1      2      3      4   */  
var lista =[ "Juan",11224, funcion(), unaVariable, true]
```

```
/*   indice:    0      1      2      3      4   */  
var lista =[true, 11224, unaVariable, funcion(), "Juan"]
```

En caso de que deseemos llamar a un valor de la lista, lo podemos ubicar sabiendo su índice: colocamos el nombre de la variable donde está almacenado el array y, entre corchetes, el índice del dato que requerimos.

EJEMPLO

```
var lista =[ "Juan",11224, funcion(), unaVariable, true]
```

```
lista[1] —> /* nos devuelve el valor 11224*/
```

2.Métodos

Para poder manipular o acceder a los valores de estos arrays, JavaScript nos provee de distintos métodos, los que podemos definir como pequeñas operaciones ya configuradas en el lenguaje. Los más utilizados en los arrays son los que detallaremos a continuación.

length():

Método que nos permite saber el largo del array, es decir que nos indica la cantidad de valores o datos que contiene el array.

Sintaxis

`nombreDelArray.length()`

`/* como resultado, nos devuelve el número de la cantidad de elementos que tiene
*/`

push():

Método que nos permite agregar un dato o valor al **final** del array; es decir, agregar un valor luego del último de la derecha.

Sintaxis

`nombreDelArray.push(dato)`

pop():

Método que nos permite eliminar el valor o dato **final** del array; es decir, el primer dato que se encuentra en el extremo derecho del array.

Sintaxis

`nombreDelArray.pop()`

shift():

Método que nos permite eliminar el valor o dato **inicial** del array; es decir, el primer dato que se encuentra en el extremo izquierdo del array.

Sintaxis

`nombreDelArray.shift()`

unshift():

Método que nos permite agregar un dato o valor al **inicio** del array; es decir, agregar un valor a la izquierda del primer valor que se encuentra en el extremo izquierdo.

`nombreDelArray.unshift(dato)`

indexOf():

Método que nos permite identificar el índice que tiene un elemento del array.

Sintaxis

`nombreDelArray.indexOf(dato)`

splice():

Método que nos permite eliminar uno o varios datos del array que no se encuentren al inicio o al final. También nos permite reemplazar elementos existentes por otros.

Sintaxis

`nombreDelArray.splice(indice,CantidadDeElementos,nuevoDato)`

Donde:

- *indice* es el número del índice donde vamos a pararnos.
- *CantidadDeElementos* es el número de elementos hacia la derecha que vamos a eliminar o reemplazar.
- *nuevoDato* es el dato que se va a colocar en reemplazo de los elementos que sacamos.

A tener en cuenta:

En caso de que solamente queramos sacar elementos, obviemos *nuevoDato* .
Es decir:

`nombreDelArray.splice(indice,CantidadDeElementos)`

slice():

Método que nos permite generar una “copia” de una parte del array ,donde le indicamos desde que dato inicial hasta que dato final copiar.

Para ello tenemos que definirlo en una variable , para poder guardar este nuevo

Sintaxis

`var nombreDelNuevoArray= nombreDelArray.slice(indiceInicial,indiceFinal)`

Donde:

- ***indiceInicial*** es el número del índice desde el que empezamos a copiar. Este elemento va a estar incluido en el nuevo array.
- ***indiceFinal*** es el número que representa al índice del elemento final en donde culmina la copia del array. Tener en cuenta que este elemento **no** va a estar incluido en el nuevo array.

1. Bibliografía utilizada y sugerida

https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/Array

<https://www.wextensible.com/temas/javascript-array/>

[#:~:text=Un%20Array%20es%20un%20vector%20en%20JavaScript,-Figura.&text=Un%20vector%20en%20programaci%C3%B3n%20inform%C3%A1tica,como%20traducci%C3%B3n%20del%20ingl%C3%A9s%20array%20.](#)

[Figura.&text=Un%20vector%20en%20programaci%C3%B3n%20inform%C3%A1tica,como%20traducci%C3%B3n%20del%20ingl%C3%A9s%20array%20.](#)

https://www.w3schools.com/js/js_arrays.asp

<https://jsparagatos.com/#arrays>