ARRAY (Arreglo)

wawiwa

Arreglo

Si queremos escribir un programa que reciba los salarios de 2000 empleados y los imprima.

¿Cómo lo hacemos?

Una opción es configurar 200 variables e inicializarlas con un salario, sin embargo esto consume tiempo. En lugar, podemos usar un arreglo.

wawiwa

Ejemplos de arreglos

```
<html>
<body>
<h1>Array example 1</h1>
<script>
    var colors = ["red", "blue", "browb"];
    console.log(colors);
    var mixedArr = ["dog", 3, "cat", 9, 12];
</script>
</body>
</html>
```

wawiwa

Arreglo – Continuación

Hasta ahora, para representar 10 salarios de empleados teníamos que configurar:
var salary1, salary2, … , salary10;

En lugar, definiremos un arreglo de variables:

```
var arr = []; // arreglo vacío
```

wawiwa

Arreglos

Algunas cosas para recordar:

1.) La referencia a los elementos de un arreglo se hace mediante su índice.

2.) El primer elemento de un arreglo siempre va a ser 0.

3.) Así vamos a referirnos a la primera variable: salary [0].

4.) El último index será [tamaño del arreglo menos 1].

5.) A la última variable nos referiremos así: salary [9].

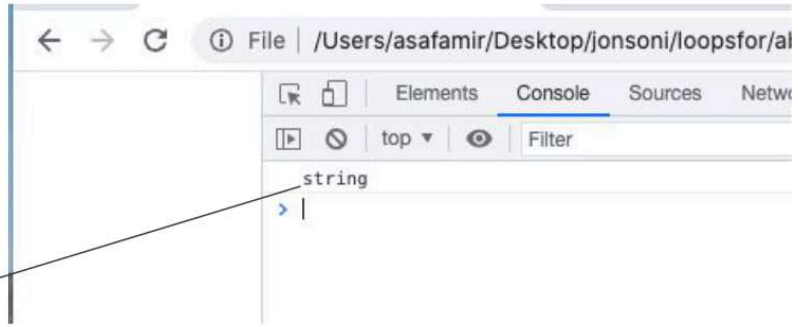
Un ejemplo de un ciclo for:

```
for (var i = 0; i < 10; i++) {
    salary[i] = 0;
}
```

wawiwa

Inicialización

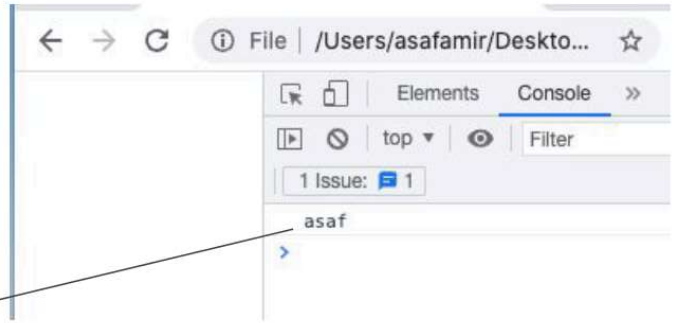
```
<html>
<body>
<script>
    var arr = [];
    arr[0] = 12;
    arr[1] = "string";
    console.log(arr[1]); // imprime string
</script>
</body>
</html>
```



wawiwa

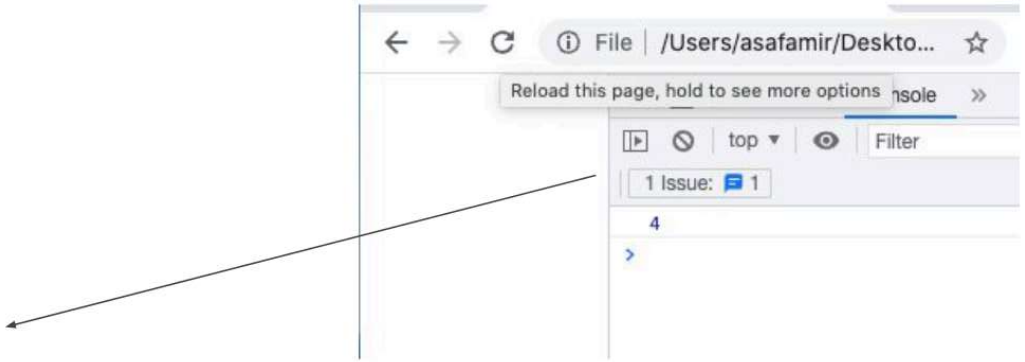
Inicialización

```
<html>
<body>
<script>
    var arr = new Array();
    arr[0] = 12;
    arr[1] = "asaf";
    console.log(arr[1]); //prime asaf
</script>
</body>
</html>
```



wawiwa

Inicialización




```
<html>
    <body>
        <script>
            var arr = [1, 4, 7, 12];
            console.log(arr[1]); //imprime 4
        </script>
    </body>
</html>
```

wawiwa

Mas de arreglos

¡No confundas el número en la línea de definición del arreglo!

var arr = [1,2,3,4,5,6,7,8,9];

Arr[8] es el último índice del arreglo

0	1	2	3	4	5	6	7	8

Propiedad - Información meta del objeto

Metodo - Funciones correspondientes al objeto.

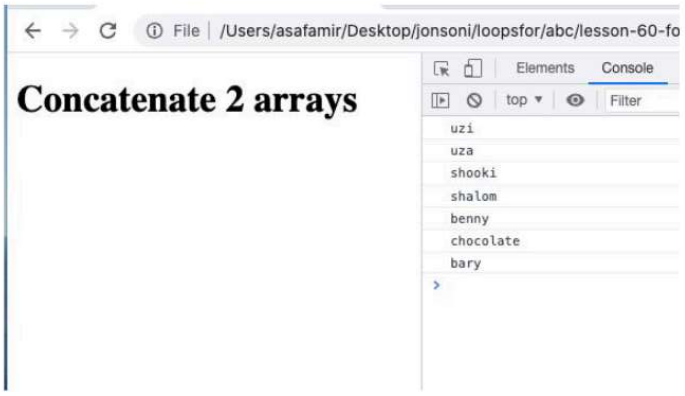
Más- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/splice#syntax

wawiwa

Concatenar

El método concat() se usa para combinar dos o más arreglos. Este método no cambia los arreglos existentes, retorna un nuevo arreglo.

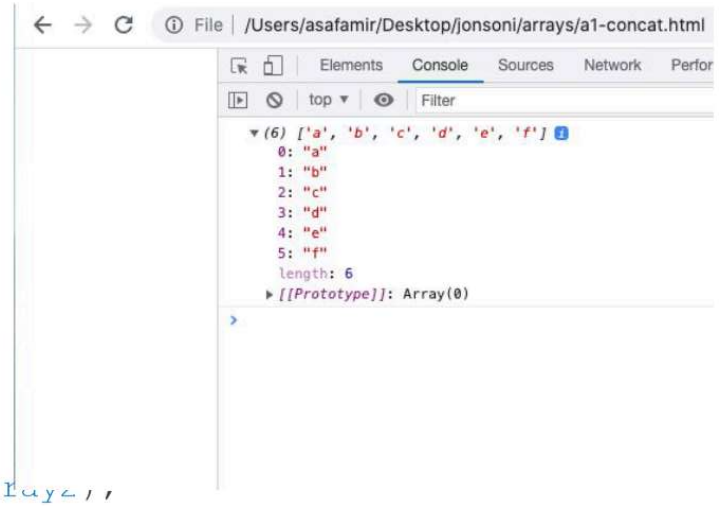
```
<!DOCTYPE html>
<html>
<body>
   <h1>Concatenate 2 arrays</h1>
   <script>
        var arr1 = new Array("uzi", "uza", "shooki");
        var arr2 = new Array("shalom", "benny", "chocolate", "bary");
        var all = arr1.concat(arr2);
        for (var i = 0; i < all.length; i++) {
            console.log(all[i]);
        }
   </script>
</body>
</html>
```



wawiwa

Concat mdn examples

```
<html>
    <body>
        <script>
            const array1 = ['a', 'b', 'c'];
            const array2 = ['d', 'e', 'f'];
            const array3 = array1.concat(array2);
            console.log(array3);
            // expected output: Array ["a", "b", "c", "d", "e", "f"]
        </script>
    </body>
</html>
```

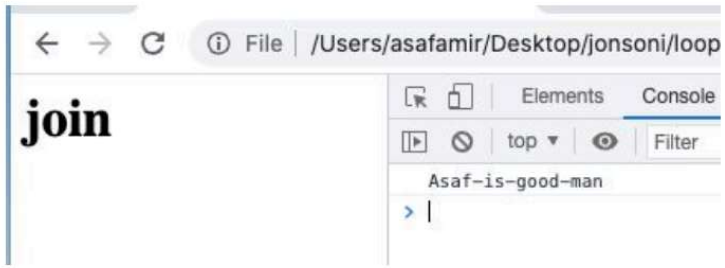


wawiwa

Join

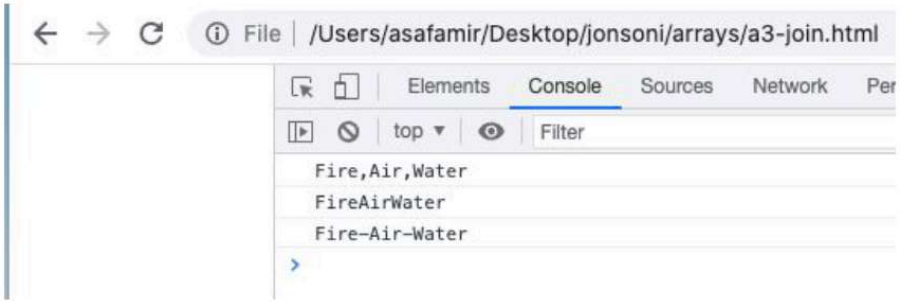
The join() method creates and returns a new string by concatenating all of the elements in an array (or an array-like object), separated by commas or a specified separator string. If the array has only one item, then that item will be returned without using the separator.

```
<html>
<body>
   <h1>join</h1>
   <script>
        var arr = new Array("Asaf", "is", "good", "man");
        var str = arr1.join("-"); //return string object :
Asaf-is-good-man
        console.log(str);
   </script>
</body>
</html>
```



wawiwa

Join mdn example

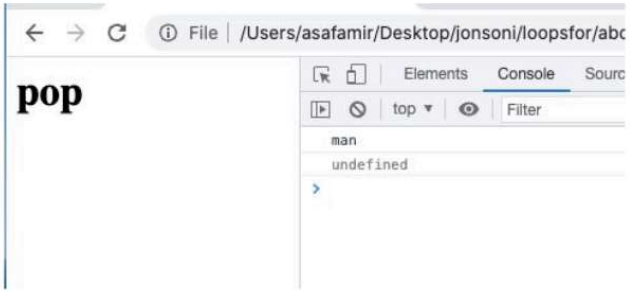



```
<html>
    <body>
        <script>
            const elements = ['Fire', 'Air', 'Water'];
            console.log(elements.join());
            // expected output: "Fire,Air,Water"
            console.log(elements.join(''));
            // expected output: "FireAirWater"
            console.log(elements.join('-'));
            // expected output: "Fire-Air-Water"
        </script>
    </body>
</html>
```

wawiwa

Pop

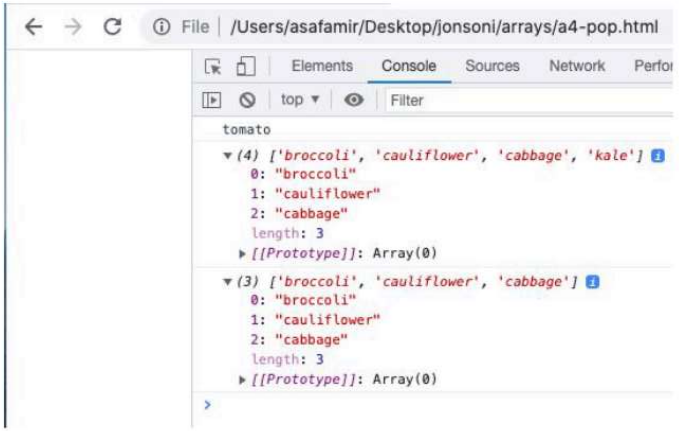
The pop() method removes the last element from an array and returns that element. This method changes the length of the array.




```
<html>
<body>
    <h1>pop</h1>
    <script>
                var arr = new Array("Asaf", "is", "good", "man");
                var strpop = arr.pop();
                console.log(strpop);
                document.write(arr[3]); // undefined
    </script>
</body>
</html>
```

wawiwa

Pop mdn examples



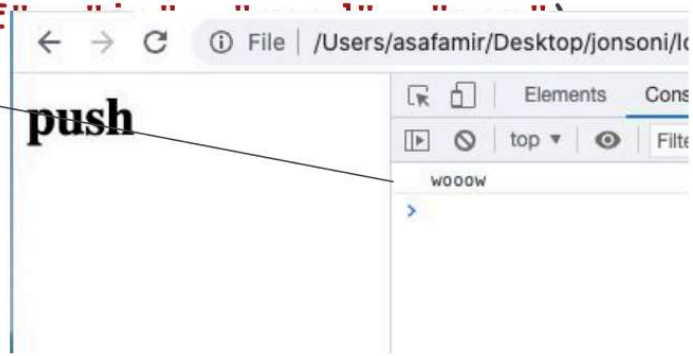

```
<html>
    <body>
        <script>
            const plants = ['broccoli', 'cauliflower', 'cabbage', 'kale', 'tomato'];
            console.log(plants.pop());
            // expected output: "tomato"
            console.log(plants);
            // expected output: Array ["broccoli", "cauliflower", "cabbage", "kale"]
            plants.pop();
            console.log(plants);
            // expected output: Array ["broccoli", "cauliflower", "cabbage"]
        </script>
    </body>
</html>
```

wawiwa

Push

The push() method adds one or more elements to the end of an array and returns the new length of the array.

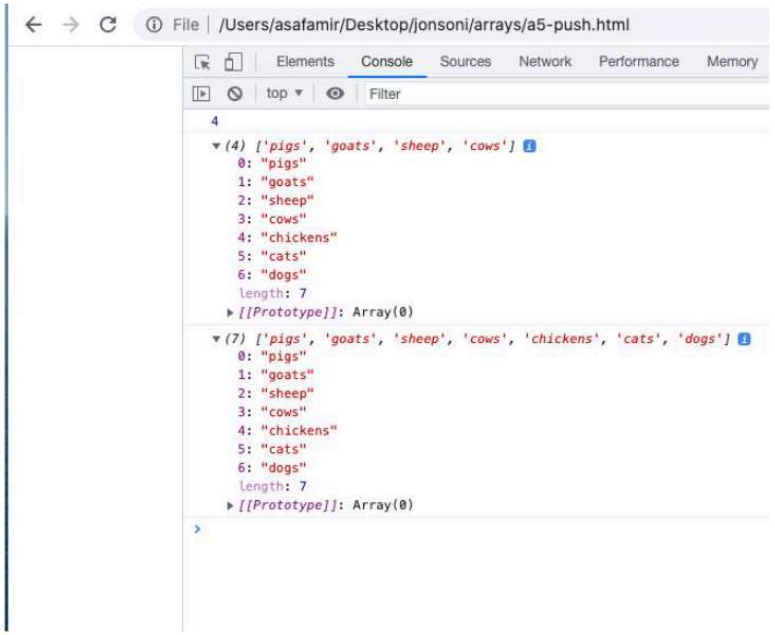
```
<html>
<body>
    <h1>push</h1>
    <script>
        var arr = new Array("Asaf"
        arr.push("wooow");
        console.log(arr[4]);
    </script>
</body>
</html>
```



wawiwa

Push mdn examples

```
<html>
    <body>
        <script>
            const animals = ['pigs', 'goats', 'sheep'];
            const count = animals.push('cows');
            console.log(count);
            // expected output: 4
            console.log(animals);
            // expected output: Array ["pigs", "goats", "sheep", "cows"]
            animals.push('chickens', 'cats', 'dogs');
            console.log(animals);
            // expected output: Array ["pigs", "goats", "sheep", "cows", "chickens",
"cats", "dogs"]
        </script>
    </body>
</html>
```



wawiwa

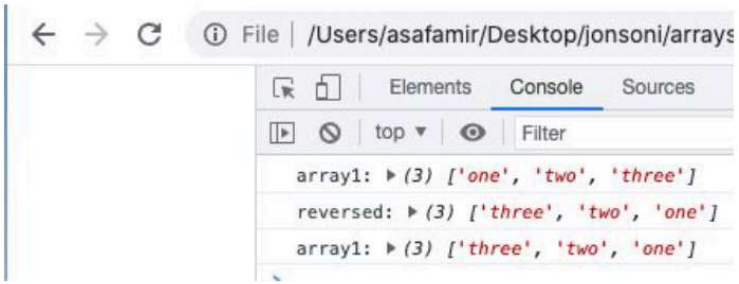
Reversa

El método reverse() method invierte un arreglo en su lugar. El primer elemento del arreglo se convierte el último y el último se convierte en el primero.

```
<html>
<body>
   <h1>reverse</h1>
   <script>
        var arr = new Array("Asaf", "is", "good", "man");
        arr.reverse();
        var strreverse = arr.join("-")
        console.log(strreverse)
   </script>
</body>
</html>
```

wawiwa

Ejemplos de reverse en mdn



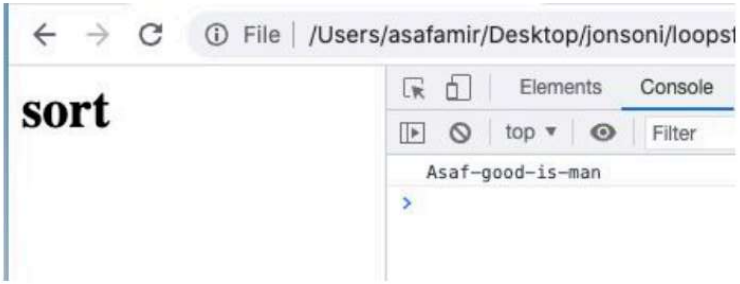

```
<html>
    <body>
        <script>
            const array1 = ['one', 'two', 'three'];
            console.log('array1:', array1);
            // resultado esperado: "array1:" Array ["one", "two", "three"]
            const reversed = array1.reverse();
            console.log('reversed:', reversed);
            // resultado esperado : "reversed:" Array ["three", "two", "one"]
            // Ten cuidado: reverse es destructivo -- cambia el arreglo original.
            console.log('array1:', array1);
            // resultado esperado : "array1:" Array ["three", "two", "one"]
        </script>
    </body>
</html>
```

wawiwa

Ordenar

El método sort() ordena los elementos de un arreglo in place y retorna el arreglo ordenado. El orden de clasificación predeterminado es ascendente, basado en la conversión de los elementos en strings y luego comparando sus secuencias de valores de unidades de código UTF-16.

```
<html>
<body>
   <h1>sort</h1>
   <script>
         var arr = new Array("Asaf", "is", "good", "man");
         arr1.sort();
         var strSort = arr.join("-");
         console.log(strSort);
   </script>
</body>
</html>
```



wawiwa

Ejemplos de sort en mdn

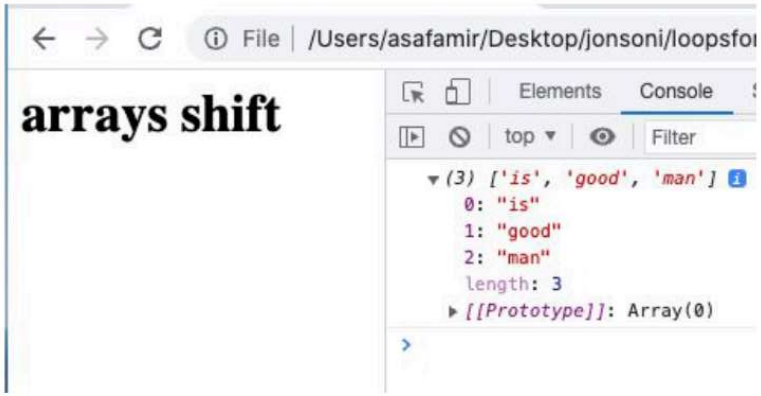

```
<html>
    <body>
        <script>
            const months = ['March', 'Jan', 'Feb', 'Dec'];
            months.sort();
            console.log(months);
            // resultado esperado: Array ["Dec", "Feb", "Jan", "March"]
            const array1 = [1, 30, 4, 21, 100000];
            array1.sort();
            console.log(array1);
            // resultado esperado: Array [1, 100000, 21, 30, 4]
        </script>
    </body>
</html>
```

wawiwa

Desplazar

El método shift() quita el primer elemento de un arreglo y retorna ese elemento. Este método cambia el largo del arreglo.

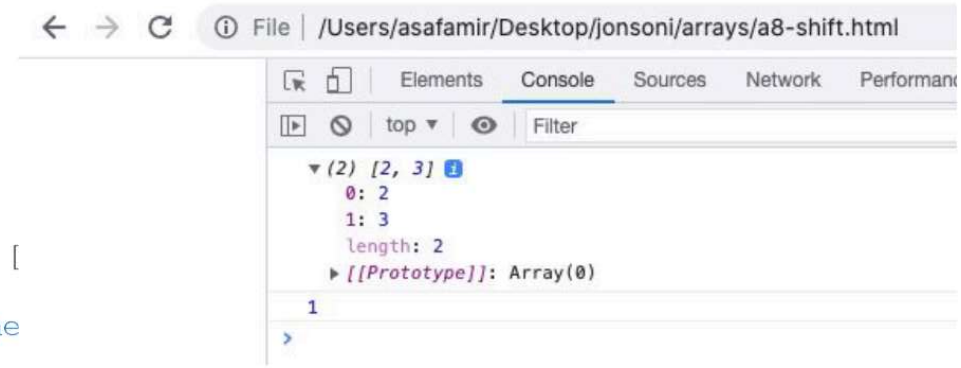
```
<html>
<body>
   <h1>arrays  shift</h1>
   <script>
         var arr = new Array("Asaf", "is", "good",
"man");
         arr.shift(arr);
         console.log(arr);
   </script>
</body>
</html>
```



wawiwa

Ejemplos de shift en mdn

```
<html>
    <body>
        <script>
            const array1 = [
            const firstEleme
            console.log(array1);
            // resultado esperado: Array [2, 3]
            console.log(firstElement);
            // resultado esperado : 1
        </script>
    </body>
</html>
```



wawiwa

IndexOf

El método indexOf() retorna el primer índice en el que se puede encontrar el elemento dado, o -1 si no está presente.

```
<html>
<body>
   <h1>arrays  indexof</h1>
    <script>
        var arr = new Array("Asaf", "is", "good", "man");
        var pos = arr.IndexOf("Asaf");
        console.log("The inde
    </script>
</body>
</html>
```

wawiwa

Ejemplos de indexOf en mdn

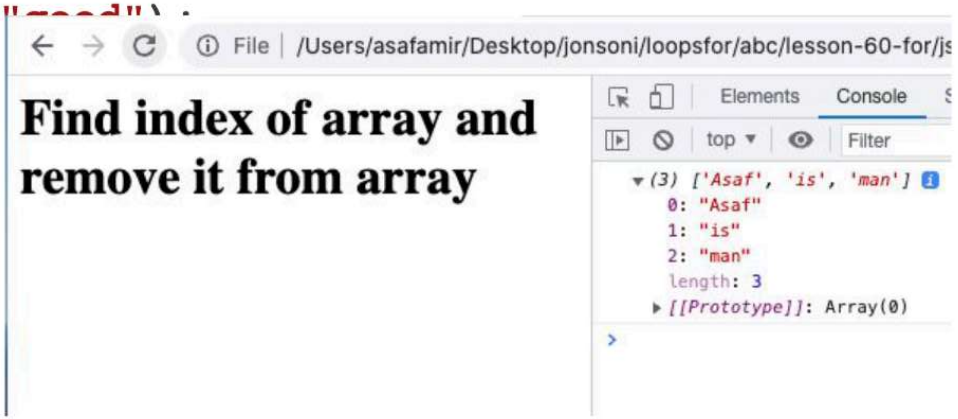
```
<html>
    <body>
        <script>
            const beasts = ['ant', 'bison', 'camel', 'duck', 'bison'];
            console.log(beasts.indexOf('bison'));
            // resultado esperado : 1
            // empieza del índice 2
            console.log(beasts.indexOf('bison', 2));
            // resultado esperado : 4
            console.log(beasts.indexOf('giraffe'));
            // resultado esperado : -1
        </script>
    </body>
</html>
```

wawiwa

Splice

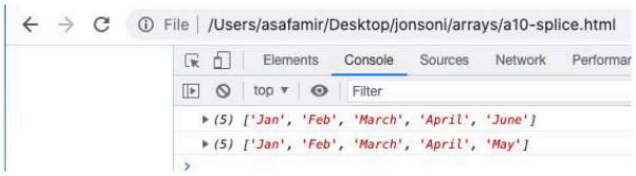
El método splice() cambia los contenidos de un arreglo al remover o reemplazar elementos existentes y/o agregando nuevos elementos en el lugar. Para acceder a una parte de un arreglo sin modificarlo, ver slice().

```
<html>
<body>
  <h1>Find index of array and remove it from array </h1>
  <script>
      var arr = new Array("Asaf", "is", "good", "man");
      var pos = arr.indexOf("good");
      arr.splice(pos, 1);
      console.log(arr);
  </script>
</body>
</html>
```



wawiwa

Ejemplo de splice en mdn




```
<html>
    <body>
        <script>
            const months = ['Jan', 'March', 'April', 'June'];
            months.splice(1, 0, 'Feb');
            // inserta en el índice 1
            console.log(months);
            // resultado esperado: Array ["Jan", "Feb", "March", "April", "June"]
            months.splice(4, 1, 'May');
            // reemplaza 1 elemento en el índice 4
            console.log(months);
            // resultado esperado: Array ["Jan", "Feb", "March", "April", "May"]
        </script>
    </body>
</html>
```

wawiwa

Cortar

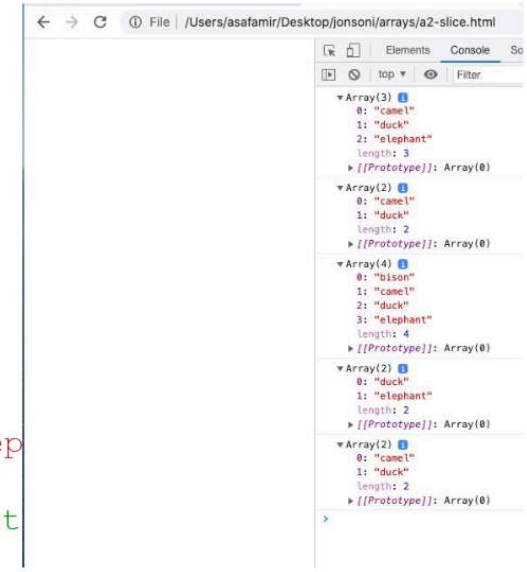
El método slice() retorna una copia superficial de una porción de un arreglo en un arreglo seleccionado de principio a fin (el fin puede no estar incluido) donde inicio y fin representan los índices de los elementos en el arreglo. El arreglo original no será modificado.

```
<html>
<body>
   <h1>Arrays slice</h1>
   <script>
       var arr = new Array("Asaf", "is", "good", "man");
       var arr = arr.slice();
       console.log(arr);
   </script>
</body>
</html>
```

wawiwa

Ejemplos de slice

```
<html>
    <body>
        <script>
            const animals = ['ant', 'bison', 'camel', 'duck', 'elep
            console.log(animals.slice(2));
            // resultado esperado: Array ["camel", "duck", "elephant
            console.log(animals.slice(2, 4));
            // resultado esperado : Array ["camel", "duck"]
            console.log(animals.slice(1, 5));
            // resultado esperado : Array ["bison", "camel", "duck", "elephant"]
            console.log(animals.slice(-2));
            // resultado esperado : Array ["duck", "elephant"]
            console.log(animals.slice(2, -1));
            // resultado esperado : Array ["camel", "duck"]
        </script>
    </body>
</html>
```

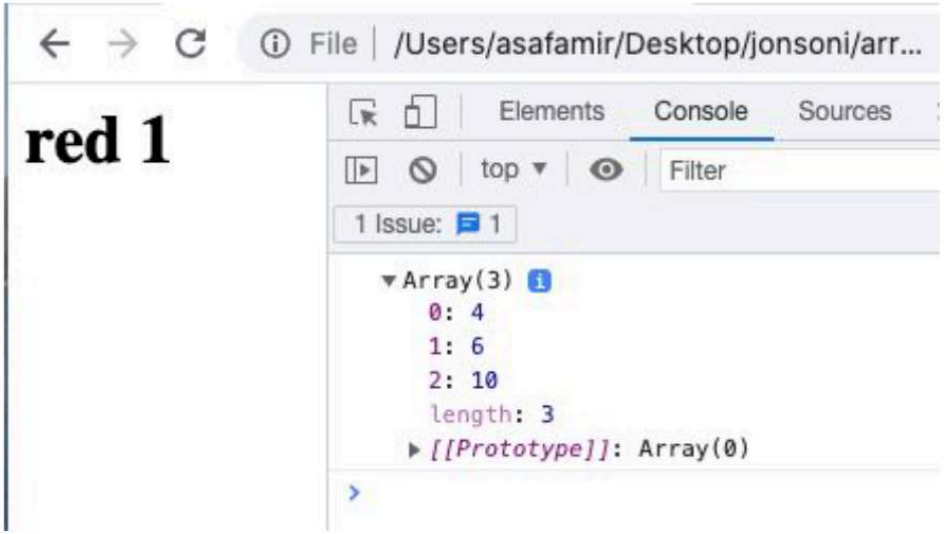


wawiwa

Hazlo tú mismo 1

Escribe un script que inicialice un arreglo con los números 4, 6, 10.

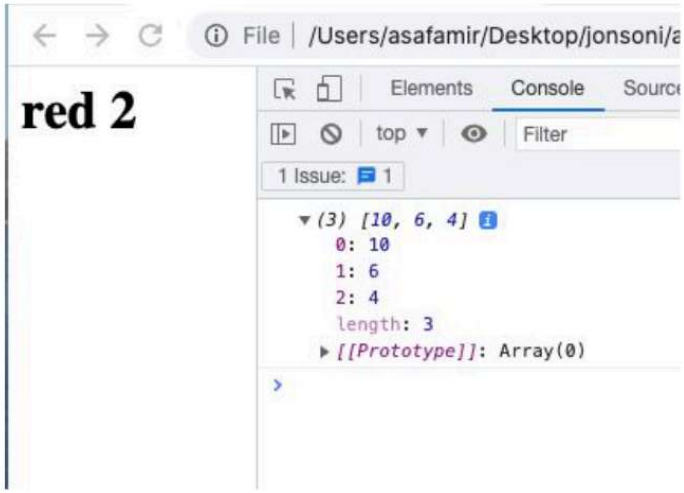
Imprime el arreglo en la consola.



wawiwa

Hazlo tú mismo 2

Escribe un script que inicialice un arreglo con los números 4, 6, 10.
Imprime el arreglo invertido en la consola.



wawiwa

Hazlo tú mismo 3

Escribe un script que inicializa un arreglo con los números 400, 600, 35.

Imprime el arreglo en la consola.

Muestra en alerta los números en orden **ascendente**.

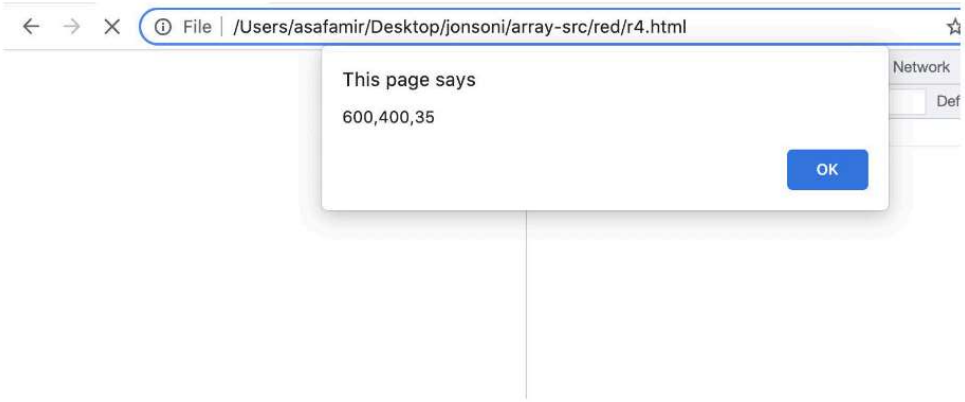
wawiwa

Hazlo tú mismo 4

Escribe un script que inicializa un arreglo con los números 400, 600, 35.

Imprime el arreglo en la consola.

Muestra en alerta los números en orden **descendente**.



wawiwa

Hazlo tú mismo 5

Escribe un script que inicializa un arreglo con los números 400, 600, 35.

Imprime el arreglo en la consola.

Borra el último elemento del arreglo y muestra en alerta los número en orden **descendente**.

wawiwa

Hazlo tú mismo 6

Escribe un script que inicialice un set con los números 1, 2, 3 e imprime el set en la consola.

wawiwa

Hazlo tú mismo 7

1. Agrega un botón con la leyenda Add numbers to array.
   a. Agrega un párrafo con id="numbers".
   b. Declara un arreglo vacío (let arr = []).
   c. Escribe un script que recibe 5 números de la ventana emergente y hazles "push" al arreglo.
   d. El script mostrará en id="numbers" el arreglo con una coma entre cada número.
      i. Hint :

```
for (var i = 0; i < arr.length; i++) {
    document.getElementById("numbers").append(arr[i] + ",");
}
```

   e. Invoca las funciones al presionar el botón.

wawiwa

¿Preguntas?