



Sesión 5 – Semana 2

Objetos y arrays

W W W . M A K A I A . O R G

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



Contenido

1. Objetos

1. Definición
2. Propiedades de un objeto
3. ¿Cómo crear objetos?
4. ¿Cómo acceder a las propiedades de los objetos?

2. Arrays

1. Definición
2. ¿Cómo crear un array?
3. ¿Cómo acceder a los elementos de los arrays?
4. Métodos de Arrays



Objetos en JavaScript

¿Qué son?

Los objetos son abstracciones inspiradas en el mundo real que permiten estructurar objetos ficticios en JavaScript de forma simple usando grupos de pares de **key/value**.

Un objeto es una colección de **propiedades**, y una **propiedad** es una asociación entre un nombre (o **key**) y un **valor**. El valor de una propiedad puede ser una **función**, en cuyo caso la propiedad es conocida como un **método**. Además de los objetos que están predefinidos en el navegador, puedes definir tus propios objetos.

W W W . M A K A I A . O R G

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



W W W . M A K A I A . O R G



Objetos en JavaScript

Un objeto es un entidad independiente con **propiedades** y tipos. Como una taza, por ejemplo. Una taza es un objeto con propiedades. Una taza tiene un color, un diseño, un peso, un material del que está hecha, etc. Del mismo modo, los objetos de JavaScript pueden tener **propiedades que definan sus características**.

W W W . M A K A I A . O R G

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



W W W . M A K A I A . O R G



Propiedades de un objeto

Una propiedad de un objeto se puede explicar como una **variable adjunta al objeto**. Las propiedades de un objeto básicamente son lo mismo que las variables comunes de JavaScript, excepto por el nexo con el objeto. Las propiedades de un objeto definen las características del objeto. Accedes a las propiedades de un objeto con una simple notación de puntos:

```
objectName.propertyName
```

W W W . M A K A I A . O R G

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



W W W . M A K A I A . O R G



¿Cómo crear nuestros propios objetos?

Un objeto se puede crear de varias manera:

1^{er} Manera:

1. Inicializando el nombre de la variable o constante donde vamos a guardar el objeto.
2. El contenido del objeto debe ir entre {}
3. Dentro de las llaves colocamos key: value, donde la key es el nombre de la propiedad y el valor puede ser de cualquier tipo de datos (strings, número, booleano, etc).

```
let myCar = {  
  make: 'Ford',  
  model: 'Mustang',  
  year: 1969  
};
```



¿Cómo crear nuestros propios objetos?

2^{da} Manera:

1. Inicializando el nombre de la variable o constante donde vamos a guardar el objeto.
2. Indicando la creación de un nuevo objeto usando `new Object()`.
3. Definiendo las propiedades, asignándole un valor.

```
let myCar = new Object();  
myCar.make = 'Ford';  
myCar.model = 'Mustang';  
myCar.year = 1969;
```

W W W . M A K A I A . O R G

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



W W W . M A K A I A . O R G



¿Cómo crear nuestros propios objetos?

2^{da} Manera:

1. Inicializando el nombre de la variable o constante donde vamos a guardar el objeto.
2. Indicando la creación de un nuevo objeto usando `new Object()`.
3. Definiendo las propiedades, asignándole un valor.

También se pueden establecer las propiedades mediante la **notación de corchetes**.

```
let myCar = new Object();  
myCar['make'] = 'Ford';  
myCar['model'] = 'Mustang';  
myCar['year'] = 1969;
```




Acceder a las propiedades de un objeto

1. La notación de punto es la que más se emplea cuando se conoce de antemano el nombre de la propiedad a la cual se quiere acceder.
1. Se puede acceder a las propiedades mediante el uso de un valor de cadena que se almacena en una variable:

```
console.log('accediendo a una  
propiedad ' + myCar.model);
```

```
let propertyName = 'make';  
myCar[propertyName] = 'Ford';  
  
propertyName = 'model';  
myCar[propertyName] = 'Mustang';
```



Arrays

¿Qué son?

Conjunto de datos ordenados por posiciones y todos asociados en una sola variable. Los datos pueden ser de cualquier tipo de dato, es decir, es posible crear una array que tenga una cadena en la primera posición, un número en el segundo, un objeto en el tercero y así sucesivamente.

W W W . M A K A I A . O R G

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



W W W . M A K A I A . O R G



¿Cómo crear arrays?

Un array se puede crear de la siguiente manera:

1^{er} Manera:

1. Inicializando el nombre de la variable o constante donde vamos a guardar el array.
2. El contenido del array debe ir entre []
3. Dentro de los corchetes colocamos los datos que queremos almacenar o.

```
let colores = ["rojo", "azul", "verde"] ;
```

W W W . M A K A I A . O R G

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



W W W . M A K A I A . O R G



¿Cómo crear arrays?

2^{da} Manera:

1. Inicializando el nombre de la variable o constante donde vamos a guardar el array.
2. Indicando la creación de una nueva lista usando `new Array()`.
3. Colocando dentro del constructor el número de posiciones para el array o los elementos que componen el array.

```
//Se crea un array sin ninguna posición
```

```
let colores = new Array();
```

```
//Se crea un array con diez posiciones
```

```
let colores = new Array(10);
```

```
//Se crea un array con tres elementos
```

```
let colores = new Array("rojo", "azul", "verde");
```

W W W . M A K A I A . O R G

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



W W W . M A K A I A . O R G



¿Cómo acceder a los elementos de un array?

los arrays cuentan con un **índice** el cual nos indica la posición de los elementos, empezando desde 0 hasta n-1 siendo n la longitud del arreglo. Para acceder a un elemento específico simplemente ponemos **[índice]** luego del nombre del array **array[índice]**.

```
let frutas = ['Manzanas', 'Peras', 'Uvas', 'Mango', 'Patilla', 'Durazno'];

//Para conocer la longitud del array
console.log('El arreglo tiene ' + frutas.length + ' posiciones');

//Para acceder al primer elemento del array
const primerElemento = frutas[0];
console.log('Este es el primer elemento del arreglo: ' + primerElemento);

//Para acceder al segundo elemento del array
const segundoElemento = frutas[1];
console.log('Este es el segundo elemento del arreglo: ' + segundoElemento);

//Para acceder al último elemento del array
const ultimoElemento = frutas[frutas.length - 1];
console.log('Este es el último elemento del arreglo: ' + ultimoElemento);
```



Métodos en los arrays

1. Recorrer un array con `.forEach()`

```
frutas.forEach(function(elemento, indice) {  
    console.log(elemento, indice);  
})
```

WWW.MAKAIA.ORG

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



WWW.MAKAIA.ORG



Métodos en los arrays

1. Recorrer un array con `.forEach()`

```
frutas.forEach(function(elemento, indice) {  
    console.log(elemento, indice);  
})
```

WWW.MAKAIA.ORG

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



WWW.MAKAIA.ORG



Métodos en los arrays

2. Aplicar cambios a cada elemento de un array con `.map()`

```
const objetosFrutas = frutas.map(function
(elemento, indice) {
  return {
    'posicion': indice,
    'fruta': elemento
  }
})

console.log(objetosFrutas);
```

WWW.MAKAIA.ORG

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



WWW.MAKAIA.ORG



Métodos en los arrays

3. Añadir un elemento al final de un Array con `.push()`

```
let nuevaLongitud = frutas.push('Naranja') // Añade "Naranja" al final  
// ["Manzana", "Banana", "Naranja"]
```

WWW.MAKAIA.ORG

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



WWW.MAKAIA.ORG



Métodos en los arrays

4. Eliminar último elemento de un Array con `.pop()`

```
let ultimo = frutas.pop() // Elimina "Naranja" del final  
// ["Manzana", "Banana"]
```

WWW.MAKAIA.ORG

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



WWW.MAKAIA.ORG



Métodos en los arrays

5. Añadir un elemento al principio de un Array con `.unshift()`

```
let nuevaLongitud = frutas.unshift('Fresa') // Añade "Fresa" al inicio  
// ["Fresa", "Manzana", "Banana"]
```

WWW.MAKAIA.ORG

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



WWW.MAKAIA.ORG



Métodos en los arrays

6. Eliminar el primer elemento de un Array con `.shift()`

```
let primero = frutas.shift() // Elimina "Fresa" del inicio  
// ["Manzana", "Banana"]
```

WWW.MAKAIA.ORG

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



WWW.MAKAIA.ORG



Métodos en los arrays

7. Encontrar el índice de un elemento del Array con . **indexOf()**

```
let pos = frutas.indexOf('Banana') // (pos) es la posición para abreviar  
// 1
```

WWW.MAKAIA.ORG

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



WWW.MAKAIA.ORG



Métodos en los arrays

8. Eliminar un único elemento mediante su posición con `.splice()`

```
let elementoEliminado = frutas.splice(pos, 1)
// ["Manzana", "Fresa"]
```

9. Eliminar varios elementos mediante su posición con `.splice()`

```
let vegetales = ['Repollo', 'Nabo', 'Rábano', 'Zanahoria']
console.log(vegetales)
// ["Repollo", "Nabo", "Rábano", "Zanahoria"]

let pos = 1, numElementos = 2

let elementosEliminados = vegetales.splice(pos, numElementos)
// ["Nabo", "Rábano"] ==> Lo que se ha guardado en "elementosEliminados"

console.log(vegetales)
// ["Repollo", "Zanahoria"] ==> Lo que actualmente tiene "vegetales"
```





Métodos en los arrays

8. Copiar un Array con `.slice()`

```
let copiaArray = vegetales.slice();  
// ["Repollo", "Zanahoria"]; ==> Copiado en "copiaArray"
```

9. Encontrar un valor del primer elemento del array que cumple con una condición con `.find()`

```
const array1 = [5, 12, 8, 130, 44];  
  
const found = array1.find(element => element > 10);  
  
console.log(found);  
// expected output: 12
```

WWW.MAKAIA.ORG

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



WWW.MAKAIA.ORG



Métodos en los arrays

10. Encontrar el índice del primer elemento del array que cumple con una condición con `.findIndex()`

```
const array1 = [5, 12, 8, 130, 44];

const indexFound = array1.findIndex(function
(elemento) {
  return elemento > 10
});

console.log(indexFound);
```



Métodos en los arrays

11. Crear nuevo array con los elementos que cumplan con una condición `.filter()`

```
const words = ['spray', 'limit', 'elite', 'exuberant',  
  'destruction', 'present'];  
  
const result = words.filter(word => word.length > 6);  
  
console.log(result);  
// expected output: Array ["exuberant", "destruction", "present"]
```



Ejercicios

1. Consultar los métodos de los arrays, identificar cómo funcionan, para qué sirven y realizar un ejemplo de cada uno (diferente a los que aparecen en las presentes diapositivas).
2. Rellena un array con los números del 1 al 10. Muéstralo por la consola. Luego, debo eliminar el arreglo de la posición 5. Luego debo solicitar un número al usuario y ubicarlo en dicha posición.
3. Solicitar nombres de personas (no controlamos nada de momento) y lo mostramos en una lista desordenada. Deberemos de guardar los nombres en un array y comprobar si existe el nombre en el array.
4. Genera un array de números aleatorios, ordenarlos y mostrarlos en una lista desordenada. **Importante: debes crear tu mismo la lista.**
5. Genera un array de números aleatorios, ordenarlos y mostrarlos en una lista desordenada. **Importante: debes crear tu mismo la lista.**
6. Desarrollar un programa que me permita solicitar 5 datos de personas, incluyendo: nombre, apellido, teléfono, correo, y la dirección debe ser en el siguiente formato: ciudad, barrio, y calle. Esta información debo almacenarla en un array de objetos. Posteriormente, debo solicitar al usuario que escriba un nombre, y mostrar por consola toda la información del usuario que tenga dicho nombre. Si el nombre no existe en el arreglo, debería mostrar un mensaje que diga: El usuario no existe.
7. Dado el array del segundo punto, iterar con un `.foreach` y mostrar el índice y el nombre de la persona en consola.

W W W . M A K A I A . O R G

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



W W W . M A K A I A . O R G



Fuentes

1. Objetos:

[https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Working with Objects](https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Working_with_Objects)

<https://yeisondaza.com/entendiendo-los-objetos-en-javascript>

<https://es.javascript.info/object>

1. Arrays:

[https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global Objects/Array#operaciones habituales](https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/Array#operaciones_habituales)

<https://devcode.la/tutoriales/manejo-de-arrays-en-javascript/>

[https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Indexed collections](https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Indexed_collections)

W W W . M A K A I A . O R G

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



W W W . M A K A I A . O R G



■ WWW.MAKAIA.ORG
Info: comunicaciones@makaia.org

Corporación MAKAlA
Medellín, Colombia
Carrera 43A – 34-155. Almacentro
Torre Norte, Oficina 701
Teléfono: (+574) 448 03 74
Móvil: (+57) 320 761 01 76

