

# Arreglos o Array = Colección de variables

Almacena secuencia de valores. se escribe como una lista de valores entre corchetes, separados por comas. Son **útiles para agrupar y manipular data.** =“ lista de compras”.

**Los elementos de un array se almacenan en propiedades.**

Entonces, cuando hacemos `myArray[0]` le estamos realmente diciendo a JavaScript que nos devuelva el valor almacenado en la propiedad `0` del array, que está definido como el primer valor de la "primera casilla" del `array`.

ej. `Var days = ["L","M","J","V","S"]`

```
days[0] = L
days[1] = M
..... etc
```

## Definiendo o cambiando los elementos de un array

### Reemplazo de elementos

Por ejemplo, para reemplazar el primer elemento del array `listOfNumbers` por el número 1, podríamos hacer lo siguiente:

```
var listOfNumbers = [2, 3, 5, 7, 11];
listOfNumbers[0] = 1;

console.log(listOfNumbers);
// → [1, 3, 5, 7, 11]
```

### Agregar elementos

```
var listOfNumbers = [2, 3, 5, 7, 11];
listOfNumbers[5] = 13;

console.log(listOfNumbers);
// → [2, 3, 5, 7, 11, 13]
```

**podemos crear un arreglo vacío y luego definir cada elemento**

```
var listOfNumbers = [ ];

listOfNumbers[0] = 2;
listOfNumbers[1] = 3;
listOfNumbers[2] = 5;
```

```
listOfNumbers[3] = 7;
listOfNumbers[4] = 11;

// imprime el arreglo en la consola

console.log(listOfNumbers);
// → [2, 3, 5, 7, 11]
```

**\* Podemos mezclar tipo de datos de un arreglo**

**Array de 2 dimensiones ( un arreglo dentro de otro arreglo) y acceder a éste**

```
var questions = [
  ['¿En cuántos países opera Laboratorio?', 3],
  ['¿Cuál es la capital de Perú?', 'Lima'],
  ['¿Cuál es baile típico Chileno?', 'Cueca'],
  ['¿Cuál es la moneda de México?', 'Peso']
];
console.log(questions[3][1]);
// → Peso
```

## Objects

**Son diferentes a los arrays por lo siguiente:**

**Los objects utilizan string en vez de índice para acceder a los elementos.** Estos strings se denominan **llaves o Keys o propiedades** y los elementos que apuntan son denominados **valores**. Estas piezas de información se llaman *pares key-value*.

En el caso de los **objects** los *pares key-value* de información pueden estar en cualquier orden, a diferencia de los arrays

Los objetos son a menudo usado para representar cosas simples pero con **múltiples características, o atributos. (como un diccionario de elementos)**.

```
var movie = {
  title: 'Back to the Future',
  director: 'Robert Zemeckis',
  stars: [ 'Michael J. Fox', 'Christopher Lloyd', 'Lea Thompson'],
  plot: 'Marty McFly, a 17-year-old high school student, is accidentally sent
30 years into the past in a time-traveling DeLorean invented by his close
friend, the maverick scientist Doc Brown.',
  oscar: 1,
  tags: [ 'Adventure', 'Comedy', 'Sci-Fi' ]
};
```

Los Keys puede ir con o sin “ ”; **Si no colocamos las comillas, los keys deben seguir las mismas reglas de nombramiento de variables**

## Accediendo información de un object

Puede ser de dos formas:

- Puede ser con [ ] pero debe ir entre “ ”
- Puede ser con ***dot notation (.)*** Si utilizamos punto el **key debe obligatoriamente cumplir con las reglas de nombres válidos**

Ejemplo :

```
console.log(movie['title']);
// → Back to the Future
```

```
console.log(movie.title);
// → Back to the Future
```

\*cuando pedimos un key que no tiene el **objeto**? JavaScript nos devuelve **undefined**.

```
var movie = {
  title: 'Back to the Future',
  director: 'Robert Zemeckis',
  stars: [ 'Michael J. Fox', 'Christopher Lloyd', 'Lea
Thompson'],
  'year of release': 1984,
  plot: 'Marty McFly, a 17-year-old high school student, is
accidentally sent 30 years into the past in a time-
traveling DeLorean invented by his close friend, the
```

```
maverick scientist Doc Brown.',
  oscar: 1,
  tags: ['Adventure', 'Comedy', 'Sci-Fi'],
  'other awards': ['best sound', ]
};

console.log(movie.stars[0]);
// → Michael J. Fox
* Para encontrar dentro del objeto la key stars, que
corresponde a un arreglo y el índice [0]
```

## Definiendo o cambiando los elementos de un object

Podemos añadir o cambiar elementos de un object mediante **dot notación**.

Ej. Un **object** vacío y se le va añadiendo y modificando elementos.

```
// creando un object vacío
var student = { };
// añadiendo elementos al object con dot notation
student.courses = ['Intro a JS', 'Intro a UX'];
student.isActive = true;
student.points = 1000;
console.log(student);
```