



JS | Arrays

Después de esta lección podrás:

1. Crear Arrays en JavaScript.
2. Manipulación de Arrays.
3. Trabajar con alguno de los métodos más conocidos del Array.

Qué son los Arrays?

Los **Arrays** son otro tipo de dato en JavaScript. Hasta ahora habíamos visto números, cadenas de texto y booleanos (verdadero o falso). Pero ¿qué pasa si en vez de guardar un número, queremos guardar una lista de números (varios números, todos en la misma variable)?

Para eso están los arrays, son **conjuntos** de datos. Por ejemplo, imaginad que queremos guardar en una variable, la edad mía y de dos amigos:

```
var edades = [33, 31, 37];
```

Pues generamos una lista de números, al final para declarar un array, basta con ponerle unos corchetes `[]` al listado.

Los datos dentro del listado pueden ser de cualquier tipo, es decir, es posible crear una array que tenga una cadena (o string) en la primera posición, un número en el segundo, un valor booleano en el tercero y así sucesivamente.

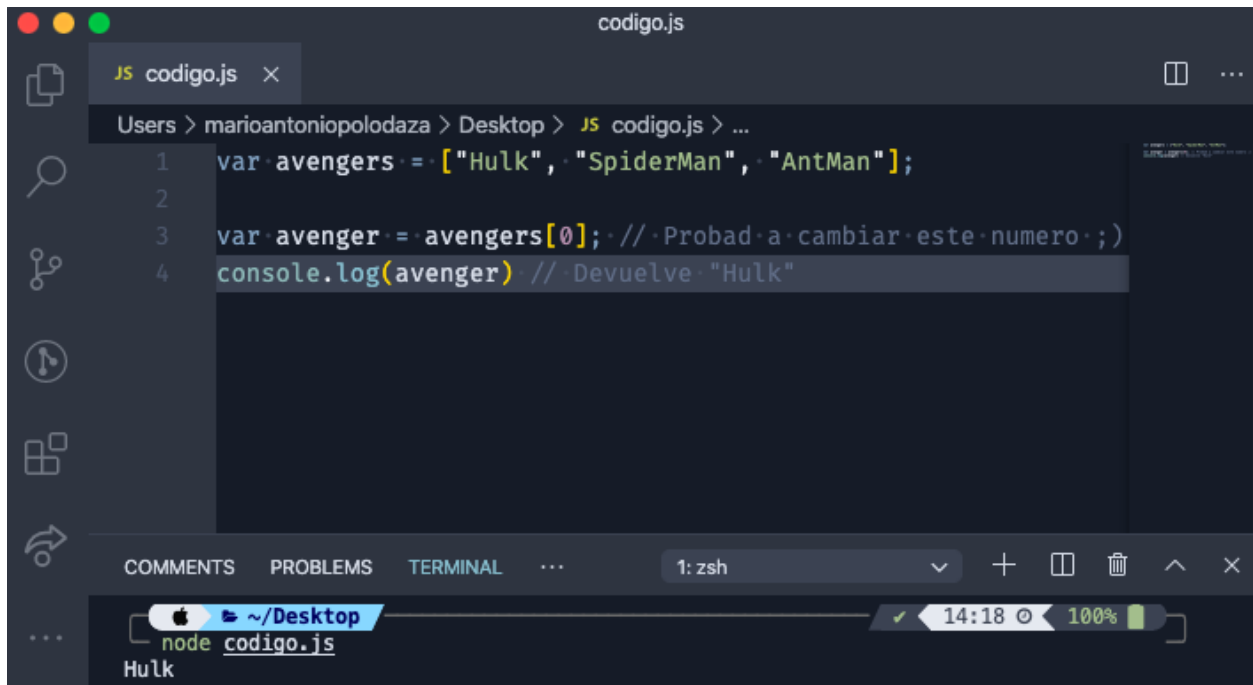
```
var ejemplo = ['string', 10, true];
```

Acceder a los elementos de un Array

Hablemos ahora de la forma en que podemos acceder los elementos del Array. Pensemos en el array como una lista de elementos, cada uno con su posición (**empezando por la 0**). Para acceder a la primera posición del array, basta con poner entre corchetes el número 0.

```
var avengers = ["Hulk", "SpiderMan", "AntMan"];

var avenger = avengers[0]; // Probad a cambiar este numero ;)
console.log(avenger) // Devuelve "Hulk"
```



```
1 var avengers = ["Hulk", "SpiderMan", "AntMan"];
2
3 var avenger = avengers[0]; // Probad a cambiar este numero;)
4 console.log(avenger) // Devuelve "Hulk"
```

COMMENTS PROBLEMS TERMINAL ... 1: zsh

node codigo.js

Hulk

Este array mostrará **Hulk**, pero modifica el número y comprueba todas las posibilidades. Recuerda que debes tener en cuenta que las posiciones empiezan **desde 0**, es decir que nuestros valores del array serán:

```
0 -> Hulk
1 -> SpiderMan
2 -> AntMan
```

Funciones de utilidad en los Arrays

Los array tienen distintas funciones de utilidad, que nos van a facilitar trabajar con ellos, algunos de ellos los detallaremos, aunque otros los iremos descubriendo a lo largo del bootcamp.

Propiedad length:

Si queremos conocer el número de elementos del array `avengers` se puede consultar mediante la propiedad `length`. Os dejamos un ejemplo:

```
avengers.length; // Devuelve 3
```

Manipular el array:

Al final el array actúa **como una pila**, que te permite añadir datos o sacarlos. Al ser una pila de elementos, siempre se añaden o eliminan por el final:

```
var avengers = ["Hulk", "Thor", "Iron-Man"];

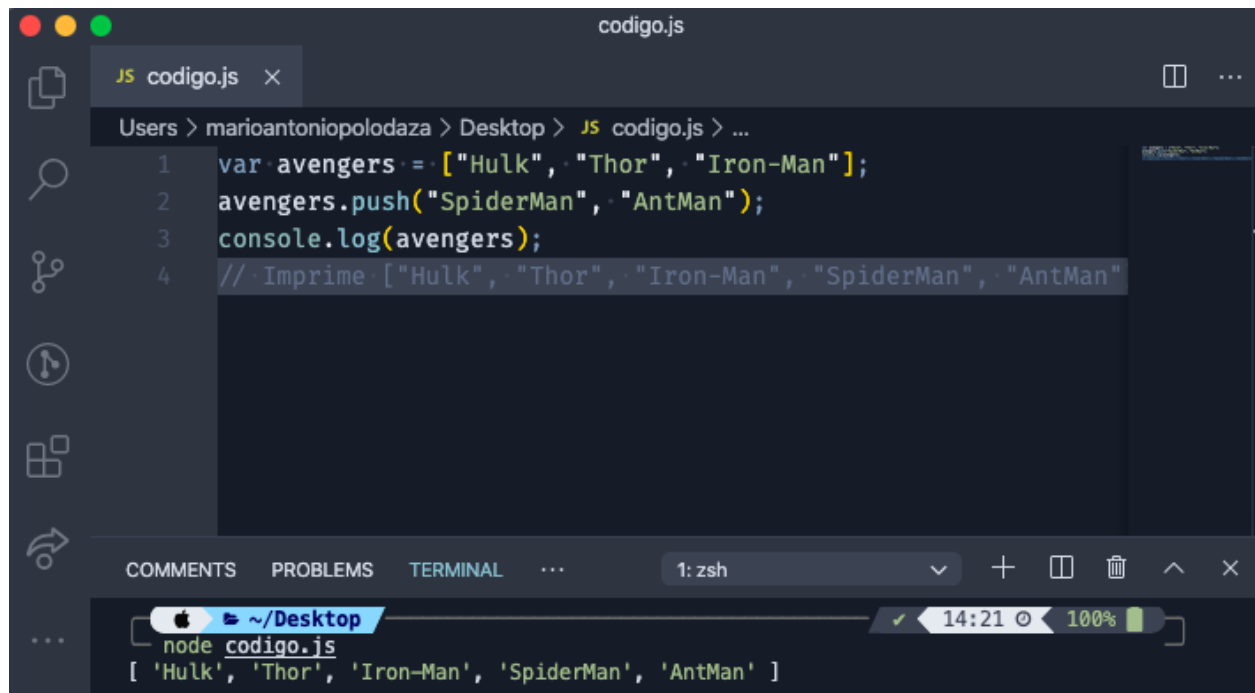
// Iron-Man  <-- Posicion: 2 (Elemento más alto de la pila)
// Thor      <-- Posicion: 1
// Hulk      <-- Posicion: 0 (Elemento más bajo de la pila)
```

La inserción de datos se puede hacer mediante el método **push** y la extracción mediante el método **pop**.

Método Push en Array:

Para añadir elementos en un array, usamos el método `push`, que inserta un nuevo elemento:

```
var avengers = ["Hulk", "Thor", "Iron-Man"];
avengers.push("SpiderMan", "AntMan");
console.log(avengers);
// Imprime ["Hulk", "Thor", "Iron-Man", "SpiderMan", "AntMan"]
```



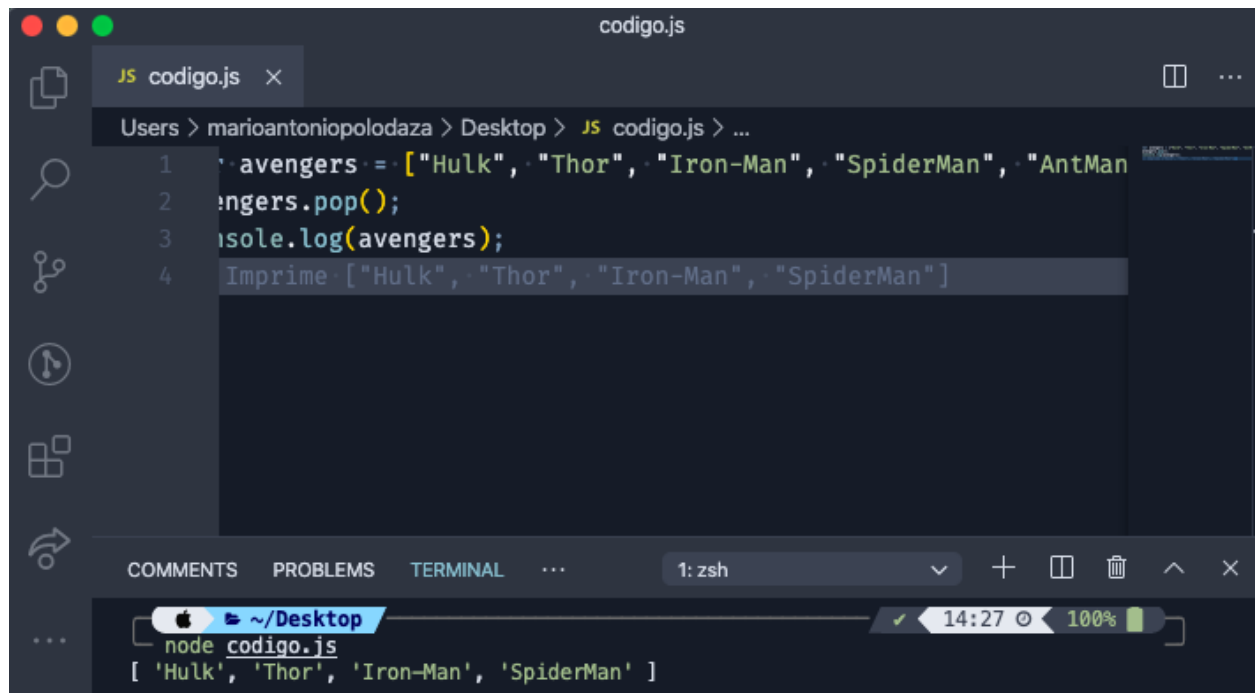
```
codigo.js
JS codigo.js x
Users > marioantoniopolodaza > Desktop > JS codigo.js > ...
1 var avengers = ["Hulk", "Thor", "Iron-Man"];
2 avengers.push("SpiderMan", "AntMan");
3 console.log(avengers);
4 // Imprime ["Hulk", "Thor", "Iron-Man", "SpiderMan", "AntMan"]

COMMENTS PROBLEMS TERMINAL ... 1: zsh
~/Desktop
node codigo.js
[ 'Hulk', 'Thor', 'Iron-Man', 'SpiderMan', 'AntMan' ]
```

Método Pop en Array:

Para eliminar elementos de un array, usamos el método `pop`, que elimina el último elemento:

```
var avengers = ["Hulk", "Thor", "Iron-Man", "SpiderMan", "AntMan"];
avengers.pop();
console.log(avengers);
// Imprime ["Hulk", "Thor", "Iron-Man", "SpiderMan"]
```



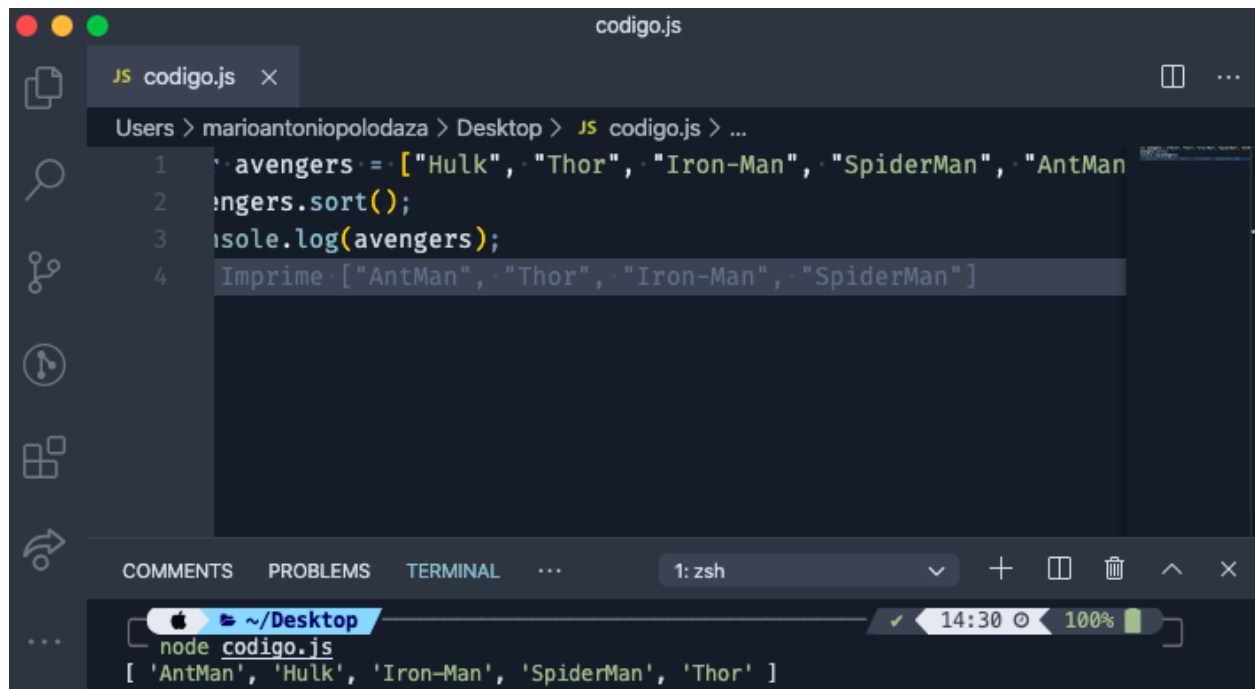
```
1 var avengers = ["Hulk", "Thor", "Iron-Man", "SpiderMan", "AntMan"]
2 avengers.pop();
3 console.log(avengers);
4 console.log(avengers);
```

```
node codigo.js
[ 'Hulk', 'Thor', 'Iron-Man', 'SpiderMan' ]
```

Método Sort:

Este método se utiliza para **ordenar** un array. En caso de que tengamos un array con elementos string, pues estos serán ordenados **alfabéticamente**:

```
var avengers = ["Hulk", "Thor", "Iron-Man", "SpiderMan", "AntMan"];
avengers.sort();
console.log(avengers);
// Imprime ['AntMan', 'Hulk', 'Iron-Man', 'SpiderMan', 'Thor']
```

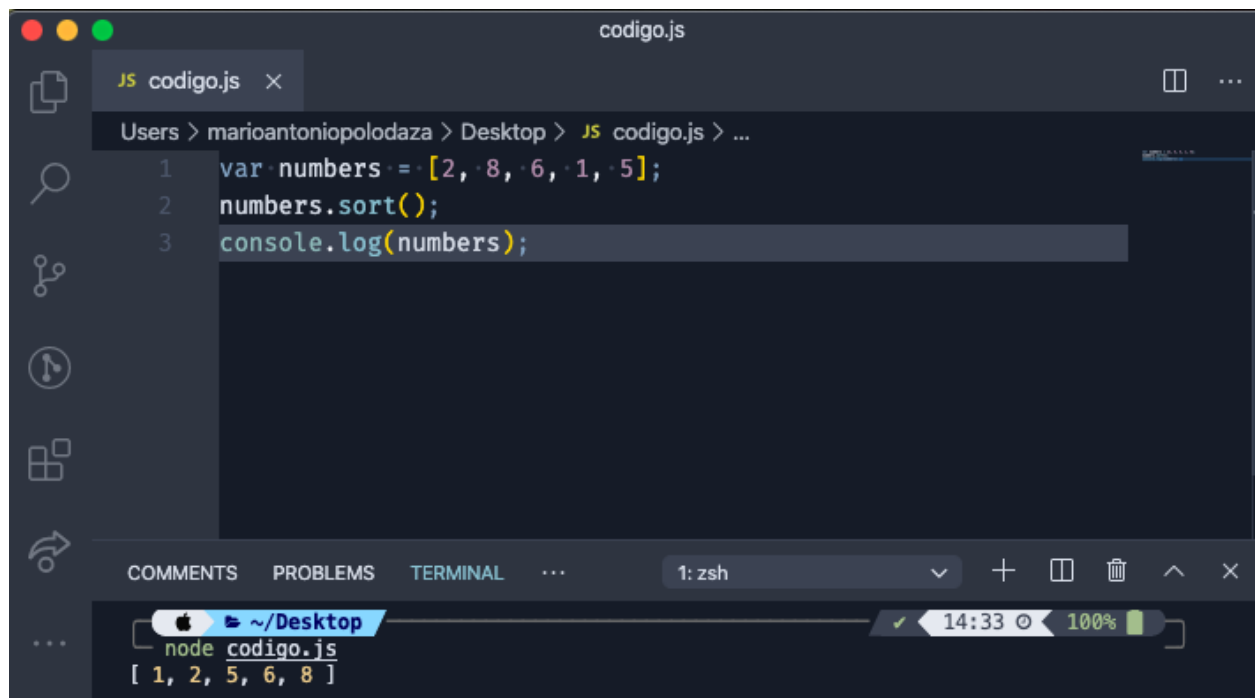


```
Users > marioantoniopolodaza > Desktop > JS codigo.js > ...  
1 avengers = ["Hulk", "Thor", "Iron-Man", "SpiderMan", "AntMan"  
2 avengers.sort();  
3 console.log(avengers);  
4 Imprime ["AntMan", "Thor", "Iron-Man", "SpiderMan"]
```

COMMENTS PROBLEMS TERMINAL ... 1: zsh

node codigo.js
['AntMan', 'Hulk', 'Iron-Man', 'SpiderMan', 'Thor']

En caso de ser números, lo ordenará de menor a mayor.



```
Users > marioantoniopolodaza > Desktop > JS codigo.js > ...  
1 var numbers = [2, 8, 6, 1, 5];  
2 numbers.sort();  
3 console.log(numbers);
```

COMMENTS PROBLEMS TERMINAL ... 1: zsh

node codigo.js
[1, 2, 5, 6, 8]

Trabajando con arrays, bucles y condicionales

Hemos visto cómo manipular un array, pero siempre desde el último elementos. Pero ¿y si queremos manipular otro dato que no sea el último?

Pues para eso os vamos a enseñar a recorrer un Array, haciendo uso de los bucles aprendidos anteriormente, en concreto un bucle **for**:

```
var avengers = ["Thor", "Spiderman", "Iron-Man", "Hulk"];

for (var i = 0; i <= avengers.length; i++) {
  console.log(avengers[i]); // Imprime los nombres de nuestros vengadores según la posición de i
}
```

Ahora introduzcamos un condicional (también aprendido en la sección anterior). Recorramos el array y imprimamos un mensaje, pero solo para **Hulk**:

```
var avengers = ["Thor", "Spiderman", "Iron-Man", "Hulk"];

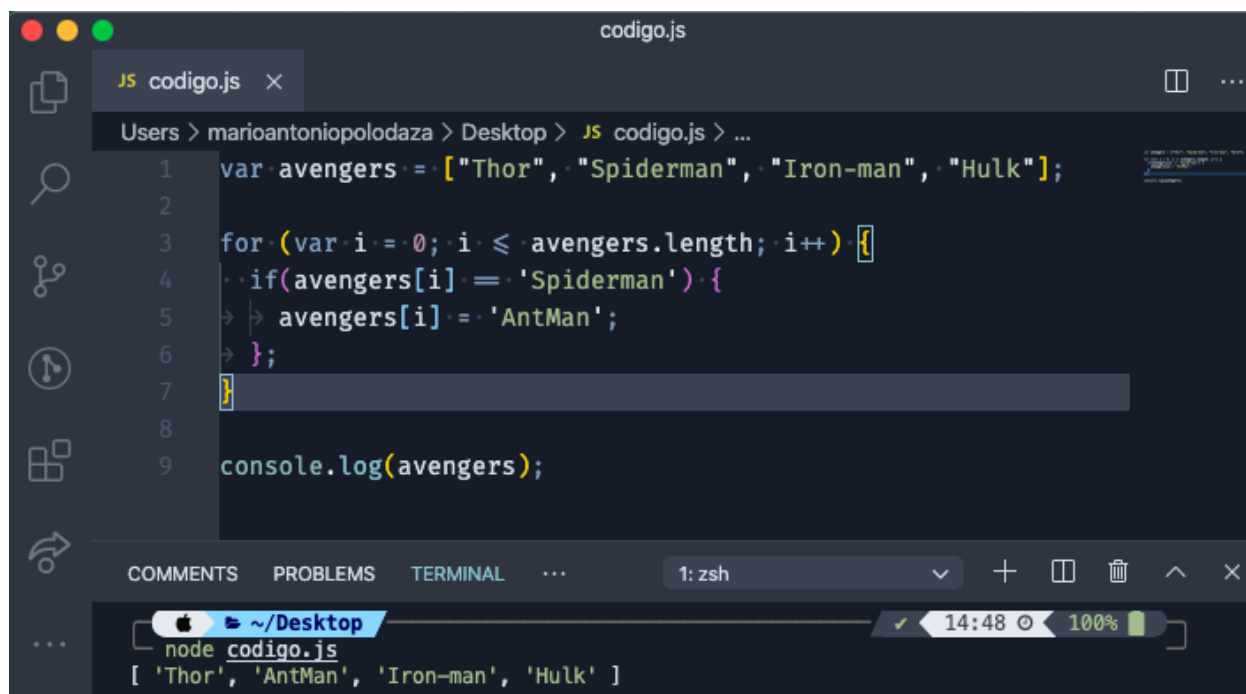
for (var i = 0; i <= avengers.length; i++) {
  if(avengers[i] == 'Hulk') {
    console.log('Este vengador se llama' + avengers[i] + ' y es el más fuerte');
  };
}
```

Y finalmente, vamos a sustituir a **Spiderman** por otro vengador:

```
var avengers = ["Thor", "Spiderman", "Iron-man", "Hulk"];

for (var i = 0; i <= avengers.length; i++) {
  if(avengers[i] == 'Spiderman') {
    avengers[i] = 'AntMan';
  };
}

console.log(avengers);
```

```
1 var avengers = ["Thor", "Spiderman", "Iron-man", "Hulk"];
2
3 for (var i = 0; i <= avengers.length; i++) {
4   if (avengers[i] === 'Spiderman') {
5     avengers[i] = 'AntMan';
6   }
7 }
8
9 console.log(avengers);
```

COMMENTS PROBLEMS TERMINAL ... 1: zsh

~/Desktop
node codigo.js
['Thor', 'AntMan', 'Iron-man', 'Hulk']

Resumen

En esta sección hemos aprendido a trabajar con listas de elementos, con arrays.

También hicimos un pequeño ejercicio completo, combinando como recorrer un array con bucles y usar un condicional para modificar valores dentro del listado.

Seguimos aprendiendo en:

 [JS | Bucles y Condicionales](#)