

Array

[Pagina principale di Javascript](#)

Un array è una struttura dati che consente di memorizzare più valori in una singola variabile. Gli array possono contenere elementi di diversi tipi di dati, come numeri, stringhe, oggetti, altri array, e così via. Gli elementi di un array sono accessibili mediante un indice, che indica la posizione dell'elemento nell'array.

Gli Array sono **variabili multidimensionali**. Un singolo elemento di un array è di fatto una variabile, al quale possiamo applicare gli operatori visti in precedenza.

Definire un array

```
//definizione di array

//definizione un elemento alla volta
let auto = []; //dobbiamo prima dire che auto è un array
auto[0]= "Fiat";
auto[1]= "Lancia";
auto[2]= "Ford";
auto[3]= "Opel";

//definizione sequenziale
let auto2 = ['fiat','lancia','ford','opel'];

//per accedere ad un singolo valore si usano gli indici
console.log(auto[3]);
console.log(auto2[1]);

//per modificare un array utilizziamo l'indice
auto[3]= "Opel";
```

Elementi in un array

Per conoscere il numero di elementi di un array possiamo ricorrere a `length`.

Array length

```
auto2 = ['fiat','lancia','ford','opel'];
//length ritorna il numero di elementi contenuti in un array
let nAuto = auto2.length //4
```

Scorrere un array

Per scorrere un array si possono usare i [cicli for](#), che verranno trattati successivamente.

scorrere un array

```
let auto2 = ['fiat','lancia','ford','opel'];
for(i=0;i<=auto2.length;i++){
  console.log(auto2[i])
}
//output
fiat
lancia
ford
opel
```

Metodi array

Metodo toString

Il metodo `toString()` in JavaScript viene utilizzato per convertire un array in una stringa. Questo metodo restituisce una stringa contenente gli elementi dell'array separati da virgole.

Metodo toString

```
let frutta = ["mela", "pera", "banana"];
let stringaFrutta = frutta.toString();
console.log(stringaFrutta); // Output: "mela,pera,banana"
```

Metodo join

Il metodo `join()` in JavaScript viene utilizzato per unire tutti gli elementi di un array in una singola stringa. Si può specificare un separatore opzionale da inserire tra gli elementi nell'output finale.

Metodo join

```
array.join(separatore)
```

- **separatore** (opzionale): Il separatore da inserire tra gli elementi nell'output finale. Se non viene specificato, di default viene utilizzata la virgola (,).

Metodo join

```
let frutta = ["mela", "pera", "banana"];
//join senza parametri è come toString()
let stringaFrutta = frutta.join();
console.log(stringaFrutta); // Output: "mela,pera,banana"
//join con parametri
let stringaFrutta = frutta.join('---');
console.log(stringaFrutta); // Output: "mela---pera---banana"
```

Metodo shift

Il metodo `shift()` in JavaScript viene utilizzato per rimuovere il primo elemento da un array e restituirlo. Questo metodo modifica l'array originale, riducendo la sua lunghezza di uno e spostando gli indici di tutti gli altri elementi verso l'inizio.

Metodo shift

```
let frutta = ["mela", "pera", "banana"];
let primoElem = frutta.shift();
console.log(primoElem);
console.log(frutta);
```

Metodo unshift

Il metodo `unshift()` in JavaScript viene utilizzato per aggiungere uno o più elementi all'inizio di un array e restituisce la nuova lunghezza dell'array.

Metodo shift

```
let frutta = ["mela", "pera", "banana"];
let lenFrutta = frutta.unshift('pesca');
console.log(lenFrutta);
console.log(frutta);
```

Metodo push()

Il metodo `push()` viene utilizzato per aggiungere uno o più elementi alla fine di un array. Modifica direttamente l'array originale e restituisce la nuova lunghezza dell'array dopo l'aggiunta degli elementi.

metodo push

```
array.push(elemento1, elemento2, ..., elementoN)
```

- **array**: L'array a cui aggiungere gli elementi.
- **elemento1, elemento2, ..., elementoN**: Gli elementi da aggiungere all'array.

metodo push

```
let numeri = [1, 2, 3];
numeri.push(4, 5);
console.log(numeri); // Output: [1, 2, 3, 4, 5]
```

Metodo pop()

Il metodo `pop()` viene utilizzato per rimuovere l'ultimo elemento da un array. Modifica direttamente l'array originale e restituisce l'elemento rimosso.

metodo pop

```
array.pop()
```

- **array**: L'array da cui rimuovere l'ultimo elemento.

metodo pop

```
let numeri = [1, 2, 3, 4, 5];  
let elementoRimosso = numeri.pop();  
console.log(elementoRimosso); // Output: 5  
console.log(numeri); // Output: [1, 2, 3, 4]
```

Metodo indexOf

Il metodo `indexOf()` restituisce l'indice della prima occorrenza di un elemento all'interno dell'array, oppure -1 se l'elemento non viene trovato.

Metodo indexOf

```
array.indexOf(elemento, indiceIniziale)
```

- **array**: L'array in cui cercare l'elemento.
- **elemento**: L'elemento da cercare all'interno dell'array.
- **indiceIniziale** (opzionale): L'indice iniziale da cui iniziare la ricerca. Se non specificato, la ricerca inizia dall'indice 0.

Metodo indexOf

```
let numeri = [1, 2, 3, 4, 5];  
let indiceTre = numeri.indexOf(3);  
console.log(indiceTre); // Output: 2 (indice dell'elemento 3)  
  
let lettere = ['a', 'b', 'c', 'd', 'e'];  
let indiceC = lettere.indexOf('c');  
console.log(indiceC); // Output: 2 (indice dell'elemento 'c')  
  
let nonTrovato = [1, 2, 3];  
let indiceSei = nonTrovato.indexOf(6);  
console.log(indiceSei); // Output: -1 (l'elemento 6 non è presente nell'array)
```

From:

<https://www.wiki.webtocom.com/> - WTC wiki

Permanent link:

<https://www.wiki.webtocom.com/doku.php?id=page:array>

Last update: **2024/04/09 10:40**

