



Coding Bootcamp

# Web fundamentals

Javascript - Parte II

## ***Programma***

**Array**

**metodi**

**reference vs valore**

**Array multi-dimensionali**

**Cicli**

### Array

L'array è una struttura dati, non primitiva( object ) che consente la memorizzazione una lista ordinata di valori.

#### Dichiarazione

```
let listChars = [ 'a', 'b', 'c', 'd' ];
```

- i valori possono essere di tutte le tipologie di dato
- possono esserci tipologie di dato differenti
- può essere vuoto
- possono esserci Array all'interno di Array
- il primo elemento è in posizione 0
- è possibile nella definizione lasciare dei valori vuoti

## Metodi per gli array

Gli array forniscono una gran quantità di metodi. Per rendere le cose più semplici, in questo capitolo li abbiamo divisi per gruppi.

### Aggiungere/rimuovere elementi

<code>arr.push()</code>	← aggiunge un elemento in coda
<code>arr.pop()</code>	← <b>estrae</b> un elemento dalla coda
<code>arr.shift()</code>	← <b>estrae</b> un elemento dalla testa
<code>arr.unshift()</code>	← aggiunge un elemento in testa
<code>arr.splice()</code>	← aggiungere e rimuovere elementi, <b>ovunque</b>
<code>arr.slice()</code>	← Ritorna un nuovo array contenente tutti gli elementi all'interno di un range di index
<code>arr.concat()</code>	← crea un nuovo array che include valori di altri array, o elementi aggiuntivi

### Metodi per gli array

#### Altri metodi utili

`arr.length` ← ritorna la lunghezza

`arr.indexOf()` ← cerca un item e ritorna l'index in cui è stato trovato, altrimenti ritorna **-1**

`arr.lastIndexOf()` ← come `indexOf` ma esegue la ricerca a partire da destra verso sinistra

`arr.includes()` ← cerca un item e ritorna **true** se lo trova

`arr.reverse()` ← inverte l'ordine degli elementi

`arr.join()` ← crea una stringa con gli elementi di `arr` incollati tra loro da un separatore

### Reference vs Valore

**let** valore = 'string';      ← Primitive      ← contiene effettivamente il **valore** assegnatole

**let** reference = [ ];      ← Object      ← contiene un **riferimento** all'oggetto

### Array multi-dimensionali

Gli array possono contenere oggetti che sono a loro volta array. Possiamo quindi utilizzare questa proprietà per creare array multi-dimensionali

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

## Cicli

I cicli hanno una grande importanza in programmazione. Grazie ad essi è possibile eseguire, per mezzo di poche righe di codice, un numero potenzialmente alto di istruzioni, che altrimenti richiederebbero, a seconda dei casi, decine, centinaia o addirittura migliaia di righe di codice.

```
let animals = ["Cat", "Dog", "Monkey"];
for (let i = 0; i < animals.length; i++) {
  console.log( animals[i] );
}
```

```
let animals = ["Cat", "Dog", "Monkey"];
for (let animal of animals) {
  console.log( animal );
}
```

```
let animals = ["Cat", "Dog", "Monkey"];
for (let key in animals) {
  console.log( animals[key] );
}
```