

# Stringhe in JavaScript

## Introduzione alle Stringhe in JavaScript

Le **stringhe** in JavaScript sono utilizzate per rappresentare e manipolare il testo. Sono un tipo di dato primitivo e possono contenere qualsiasi sequenza di caratteri, compresi numeri e simboli speciali.

Le stringhe possono essere utilizzate per:

- Archiviare e manipolare **nomi, indirizzi, numeri di telefono, codici prodotto**
- Formattare testi per output su interfaccia utente
- Elaborare input utente e validare dati

Esempi di creazione di stringhe:

```
let greeting = "Hello, World!"; // Doppie virgolette
let username = 'JohnDoe';      // Singole virgolette
let sentence = `This is a template string`; // Backticks (Template Literal)
```

Le **template strings** (o **template literals**) con i **backticks** ( ``` ) permettono di scrivere stringhe multilinea e interpolare variabili:

```
let name = "Alice";
let age = 30;
console.log(`Mi chiamo ${name} e ho ${age} anni.`);
```

---

## Caratteristiche delle Stringhe

### 1. Indicizzazione e Accesso ai Caratteri

Le stringhe in JavaScript sono **indicizzate a zero**, il che significa che il primo carattere ha indice `0`, il secondo `1` e così via.

```
let str = "JavaScript";
console.log(str[0]); // 'J'
console.log(str.charAt(4)); // 'S'
```

❌ **Sconsigliato:** Usare le parentesi quadre []

✅ **Consigliato:** Usare `charAt()` per la compatibilità con browser vecchi

```
let word = "tree";  
console.log(word[8]); // undefined  
console.log(word.charAt(8)); // '' (stringa vuota)
```

---

## 2. Immutabilità delle Stringhe

Le stringhe in JavaScript **sono immutabili**, il che significa che non possono essere modificate dopo la loro creazione.

Qualsiasi operazione che sembra modificarle, in realtà crea una nuova stringa.

```
let str = "hello";  
str[0] = "H"; // Non ha effetto  
console.log(str); // 'hello'
```

Per modificare una stringa, bisogna **creare una nuova stringa**:

```
let original = "hello";  
let modified = original.toUpperCase();  
console.log(original); // 'hello'  
console.log(modified); // 'HELLO'
```

---

## Metodi e Proprietà delle Stringhe

### 1. Proprietà `length`

Restituisce la lunghezza della stringa (numero di caratteri):

```
let alphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";  
console.log(alphabet.length); // 26
```

---

### 2. Estrazione di Parti di Stringa

JavaScript offre tre metodi principali per estrarre parti di una stringa:

Metodo	Descrizione	Sintassi
<code>slice()</code>	Estrae una porzione della stringa specificando gli indici di inizio e fine	<code>string.slice(start, end)</code>
<code>substring()</code>	Simile a <code>slice()</code> , ma non accetta indici negativi	<code>string.substring(start, end)</code>
<code>substr()</code>	Estrae una parte della stringa specificando l'indice di inizio e la lunghezza	<code>string.substr(start, length)</code>

### Esempio:

```
let text = "JavaScript is amazing";
console.log(text.slice(0, 10)); // 'JavaScript'
console.log(text.substring(0, 10)); // 'JavaScript'
console.log(text.substr(11, 7)); // 'is amaz'
```

---

## 3. Conversione Maiuscole/Minuscole

- `toUpperCase()` → Converte tutto in maiuscolo
- `toLowerCase()` → Converte tutto in minuscolo

```
let text = "JavaScript";
console.log(text.toUpperCase()); // 'JAVASCRIPT'
console.log(text.toLowerCase()); // 'javascript'
```

---

## 4. Ricerca di Testo nelle Stringhe

Metodo	Descrizione	Esempio
<code>indexOf()</code>	Restituisce l'indice della prima occorrenza di una sottostringa	<code>"hello world".indexOf("world")</code> // 6

Metodo	Descrizione	Esempio
<code>lastIndexOf()</code>	Restituisce l'indice dell'ultima occorrenza	<code>"hello hello".lastIndexOf("hello")</code> <code>// 6</code>
<code>includes()</code>	Restituisce <code>true</code> se la stringa contiene la sottostringa	<code>"hello".includes("he")</code> <code>// true</code>
<code>startsWith()</code>	Controlla se la stringa inizia con una sottostringa	<code>"JavaScript".startsWith("Java")</code> <code>// true</code>
<code>endsWith()</code>	Controlla se la stringa termina con una sottostringa	<code>"script".endsWith("ipt")</code> <code>// true</code>

---

## 5. Modifica delle Stringhe

- `replace(old, new)` → Sostituisce una sottostringa
- `trim()` → Rimuove spazi iniziali e finali
- `concat()` → Concatena più stringhe

```
let str = " Hello, JavaScript! ";
console.log(str.trim()); // 'Hello, JavaScript!'

let message = "I love Python";
console.log(message.replace("Python", "JavaScript")); // 'I love JavaScript'
```

---

## Conclusione

Le stringhe sono una parte fondamentale della programmazione JavaScript. La loro manipolazione è essenziale per la gestione di dati testuali, input utente e output su interfacce web.

### Risorse utili:

- [MDN JavaScript Strings](#)
- [Regex101 - Testa le tue espressioni regolari](#)