

Stringhe

[Pagina principale di Javascript](#)

Le stringhe in JavaScript sono delimitate da **“virgolette”** o **‘apici’**, l'utilizzo di uno o l'altro è indifferente. Se vogliamo inserire delle virgolette all'interno di una stringa delimitate dalle virgolette bisogna usare il carattere di escape \. Vale lo stesso discorso per le stringhe delimitate da apici.

Apici e virgolette

```
str = 'Stringa 1'
str2 = "Stringa 2"
strEscapeApici = 'Viva l\'italia'
strEscapeVirgolette = " viva le \"viroglette\" "
strApiciInVirgolette = " Gli ' non creano problemi nelle stringhe con virgolette "
strVirgoletteInApici = '<a href="pag2.php">Link</a>' //comodo per i tag
```

Lunghezza di una stringa

E'possibile consocere il numero di cartteri di una stringa usando .length

lunghezza stringa

```
str = "1234"
let lunghezzaStringa = str.length ;
console.log(lunghezzaStringa)
```

Concatenatore delle stringhe

Le stringe si possono concatenare, cioè aggiungere una all'altra con l'operatore +.

lunghezza stringa e concatenazione

```
str = "1234"
str1 = "5678"
str2 = str + str1
console.log(str2)

cm = 183 //è un numero
altezza = "Sono alto " + cm + " centimetri" //cm viene trattato come una stringa
console.log(altezza)
```

Metodi sulle stringhe

Vediamo alcuni metodi sulle stringhe:

Metodo Split

In JavaScript, il metodo `split()` viene utilizzato per dividere una stringa in un array di sottostringhe, utilizzando il separatore specificato.

javascript Copy code

il metodo split

```
stringa.split(separatore, limiteOpzionale)
```

- **stringa**: La stringa da dividere.
- **separatore**: Il separatore utilizzato per dividere la stringa. Può essere un carattere o una stringa. Se il separatore viene omissso, la stringa verrà divisa in un array di singoli caratteri.
- **limiteOpzionale** (opzionale): Un numero intero che specifica il numero massimo di elementi nell'array restituito.

esempio di split

```
let testo = "Ciao, come stai?";
let parole = testo.split(" ");
console.log(parole); // Output: ["Ciao,", "come", "stai?"]

let data = "2022-01-01";
let partiData = data.split("-");
console.log(partiData); // Output: ["2022", "01", "01"]
```

Metodo Slice

il metodo `slice()` viene utilizzato per estrarre una parte di una stringa e restituirne una nuova stringa. Questo metodo accetta due parametri: l'indice di inizio e l'indice di fine della porzione della stringa che si desidera estrarre.

Ecco la sintassi di base del metodo `slice()`:

Il metodo slice

```
stringa.slice(indiceInizio, [indiceFine]);
```

- **indiceInizio**: L'indice iniziale da cui iniziare l'estrazione. Se questo parametro è negativo, rappresenta un offset dalla fine della stringa. Se omissso, l'estrazione inizia dall'inizio della stringa.
- **indiceFine** (opzionale): L'indice finale fino al quale eseguire l'estrazione (ma senza includerlo). Se omissso, l'estrazione si estende fino alla fine della stringa. Se questo parametro è negativo, rappresenta un offset dalla fine della stringa.

Il metodo slice

```
let stringa = "Hello, world!";  
let substr = stringa.slice(7, 12); // Restituisce "world"  
console.log(substr);  
  
let parteIniziale = stringa.slice(0, 5); // Restituisce "Hello"  
console.log(parteIniziale);  
  
let parteFinale = stringa.slice(-6); // Restituisce "world!"  
console.log(parteFinale);
```

Metodo Replace

Il metodo `replace()` viene utilizzato per sostituire parti di una stringa con un'altra stringa. Ecco la sintassi di base del metodo `replace()`:

Il metodo replace

```
stringa.replace(sottostringa, nuovaStringa)
```

- **sottostringa**: La sottostringa o l'espressione regolare da cercare nella stringa.
- **nuovaStringa**: La stringa con cui sostituire le occorrenze della sottostringa.

Il metodo replace

```
let stringaOriginale = "La mela è rossa. La mela è dolce.";  
let nuovaStringa = stringaOriginale.replace("mela", "banana");  
console.log(nuovaStringa); // Output: "La banana è rossa. La mela è dolce."  
//è case sensitive  
let stringaOriginale2 = "La mela è rossa. La mela è dolce.";  
let nuovaStringa2 = stringaOriginale.replace("Mela", "banana");  
console.log(nuovaStringa2); // Output: "La mela è rossa. La mela è dolce."
```

Metodo indexOf

Il metodo `indexOf()` restituisce l'indice della prima occorrenza di una sottostringa all'interno di una stringa, oppure -1 se la sottostringa non viene trovata.

Il metodo indexOf

```
stringa.indexOf(sottostringa, indiceIniziale)
```

- **stringa**: La stringa in cui cercare la sottostringa.
- **sottostringa**: La sottostringa da cercare all'interno della stringa.
- **indiceIniziale** (opzionale): L'indice iniziale da cui iniziare la ricerca. Se non specificato, la ricerca inizia dall'inizio della stringa.

Il metodo indexOf

```
let testo = "Ciao, mondo!";
let indice = testo.indexOf("mondo");
console.log(indice); // Output: 6

let parola = "banana";
let indiceB = parola.indexOf("b");
console.log(indiceB); // Output: 0

let nonTrovato = "hello";
let indiceNonTrovato = nonTrovato.indexOf("world");
console.log(indiceNonTrovato); // Output: -1
```

Minuscole e Maiuscole

I metodi `toLowerCase()` e `toUpperCase()`, trasformano tutti i caratteri di una stringa in minuscolo o maiuscolo,

Maisucole e Minuscole

```
testo = "AbCdEfGh"
//minuscolo
let minuscolo = testo.toLowerCase();
console.log(minuscolo);
//maiuscolo
let maiuscolo = testo.toUpperCase();
console.log(maiuscolo);
```

Scorrere una stringa un carattere alla volta

Una stringa si può scorrere come se fosse un [array](#) di caratteri utilizzando un ciclo [for](#).

scorrere una stringa

```
let testo = "Ciao, mondo!";
for (let i = 0; i < testo.length; i++) {
  console.log(testo[i]);
}
```

From:

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

Permanent link:

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

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

