

 Cập nhật tháng 8 năm 2024

[Bài Đọc] Phương thức làm việc với string (String methods)

Trong JavaScript, chuỗi là một trong những kiểu dữ liệu quan trọng và thường được sử dụng. Để làm việc hiệu quả với chuỗi, JavaScript cung cấp nhiều phương thức (methods) tích hợp. Dưới đây là tổng quan về một số phương thức phổ biến nhất.

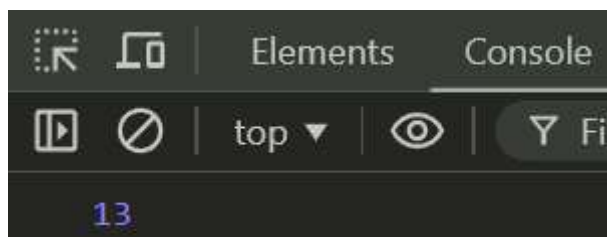
1. length

- **Mô tả:** Thuộc tính dùng để trả về độ dài (số ký tự) của chuỗi.

Ví dụ:

```
let str = 'Hello, world!';  
console.log(...data: str.length); // Output: 13
```

Kết quả:



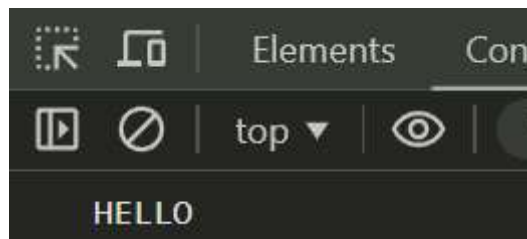
2. toUpperCase()

- **Mô tả:** Chuyển tất cả các ký tự trong chuỗi thành chữ in hoa.

Ví dụ:

```
let str = 'hello';  
console.log(...data: str.toUpperCase()); // Output: "HELLO"
```

Kết quả:



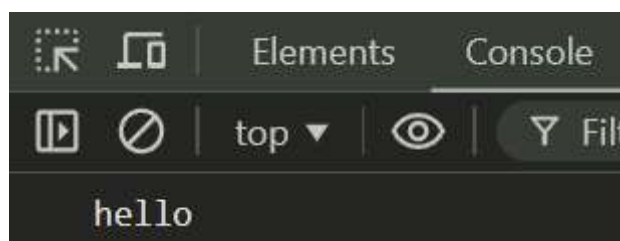
3. toLowerCase()

- **Mô tả:** Chuyển tất cả các ký tự trong chuỗi thành chữ in thường.

Ví dụ:

```
let str = 'HELLO';  
console.log(...data: str.toLowerCase()); // Output: "hello"
```

Kết quả:



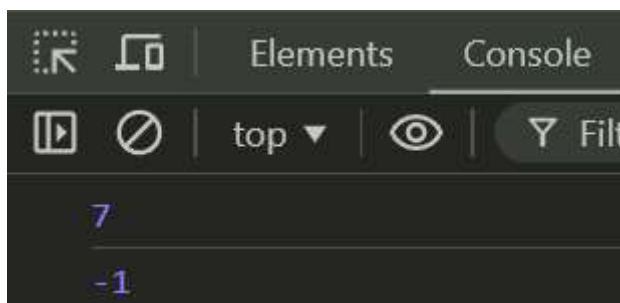
4. indexOf()

- **Mô tả:** Trả về vị trí xuất hiện đầu tiên của một chuỗi con trong chuỗi chính. Nếu không tìm thấy, trả về -1.

Ví dụ:

```
let str = "Hello, world!";  
console.log(...data: str.indexOf(searchString: "world")); // Output: 7  
console.log(...data: str.indexOf(searchString: "JavaScript")); // Output: -1
```

Kết quả:



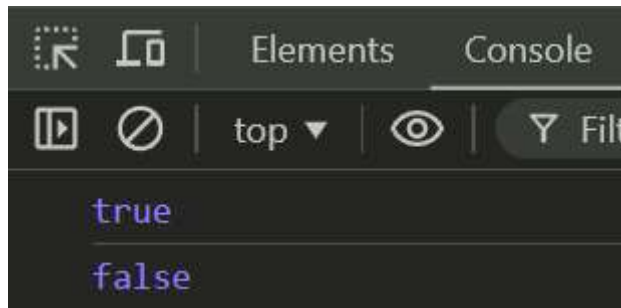
5. includes()

- **Mô tả:** Kiểm tra xem chuỗi có chứa chuỗi con được chỉ định hay không. Trả về **true** hoặc **false**.

Ví dụ:

```
let str = "Hello, world!";  
console.log(...data: str.includes(searchString: "world")); // Output: true  
console.log(...data: str.includes(searchString: "JavaScript")); // Output: false
```

Kết quả:



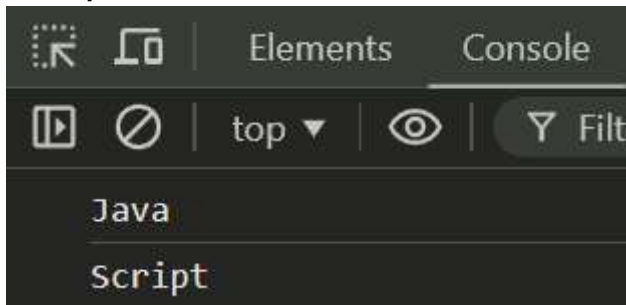
6. slice(start,end)

- Mô tả: Trích xuất một phần của chuỗi từ vị trí **start** đến **end** (không bao gồm **end**).

Ví dụ:

```
let str = "JavaScript";  
console.log(...data: str.slice(start: 0, end: 4)); // Output: "Java"  
console.log(...data: str.slice(start: 4)); // Output: "Script"
```

Kết quả:



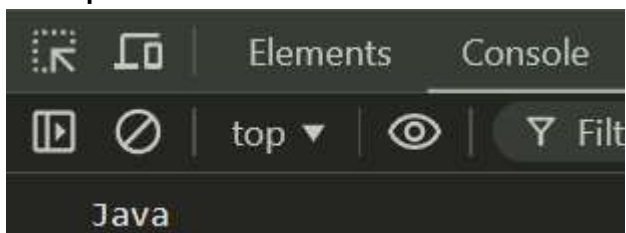
7. substring(start,end)

- Mô tả: Tương tự như `.slice()`, nhưng không hỗ trợ số âm.

Ví dụ:

```
let str = "JavaScript";  
console.log(...data: str.substring(start: 0, end: 4)); // Output: "Java"
```

Kết quả:



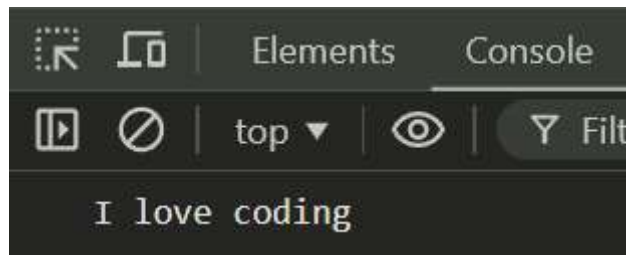
8. replace(searchValue,newValue)

- **Mô tả:** Thay thế chuỗi con đầu tiên tìm thấy bằng chuỗi mới.

Ví dụ:

```
let str = "I love JavaScript";  
console.log(...data: str.replace(searchValue: "JavaScript", replaceValue: "coding")); // Output: "I love coding"
```

Kết quả:



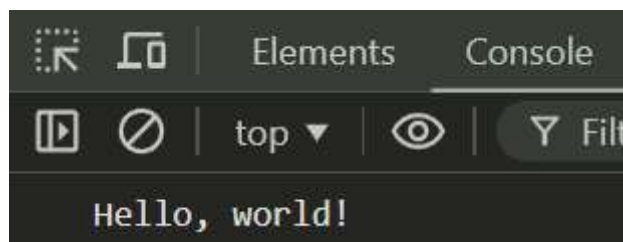
9. trim()

- **Mô tả:** Xóa bỏ khoảng trắng ở đầu và cuối chuỗi.

Ví dụ:

```
let str = " Hello, world! ";  
console.log(...data: str.trim()); // Output: "Hello, world!"
```

Kết quả:



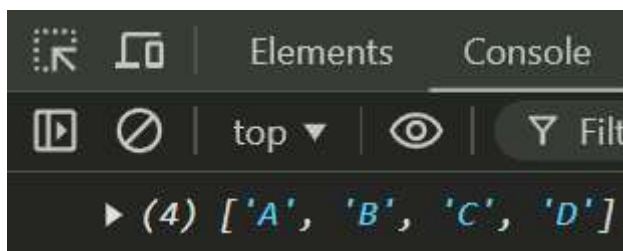
10. split(separator)

- **Mô tả:** Tách chuỗi thành một mảng các phần tử dựa trên **separator**.

Ví dụ:

```
let str = 'A, B, C, D';  
console.log(...data: str.split(separator: ', ')); // Output: ["A", "B", "C", "D"]
```

Kết quả:



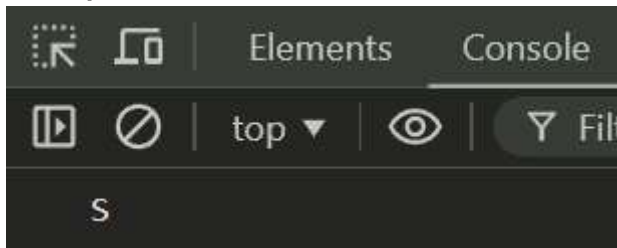
11. charAt(index)

- **Mô tả:** Trả về ký tự tại vị trí **index** trong chuỗi.

Ví dụ:

```
let str = 'JavaScript';  
console.log(...data: str.charAt(pos: 4)); // Output: "S"
```

Kết quả:



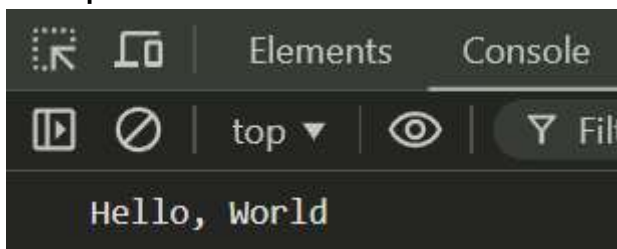
12. concat()

- Mô tả: Nối các chuỗi lại với nhau.

Ví dụ:

```
let str1 = "Hello";  
let str2 = "World";  
console.log(...data: str1.concat(...strings: ", ", str2)); // Output: "Hello, World"
```

Kết quả:



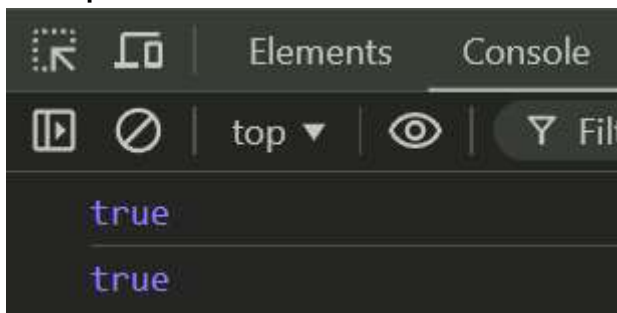
13. startsWith() và endsWith()

- Mô tả:
 - .startsWith(): Kiểm tra chuỗi có bắt đầu bằng một chuỗi con không.
 - .endsWith(): Kiểm tra chuỗi có kết thúc bằng một chuỗi con không.

Ví dụ:

```
let str = 'JavaScript';  
console.log(...data: str.startsWith(searchString: 'Java')); // Output: true  
console.log(...data: str.endsWith(searchString: 'Script')); // Output: true
```

Kết quả:



14. Tài liệu tham khảo

https://www.w3schools.com/js/js_string_methods.asp

Danh sách các bài học

