

Bài: Kiểu dữ liệu chuỗi trong JavaScript (Phần 3) - Các phương thức với chuỗi trong Javascript

Xem bài học trên website để ủng hộ Kteam: [Kiểu dữ liệu chuỗi trong JavaScript \(Phần 3\) - Các phương thức với chuỗi trong Javascript](#)

Mọi vấn đề về lỗi website làm ảnh hưởng đến bạn hoặc thắc mắc, mong muốn khóa học mới, nhằm hỗ trợ cải thiện Website. Các bạn vui lòng phản hồi đến Fanpage [How Kteam](#) nhé!

Dẫn nhập

Ở các bài trước, chúng ta đã được tìm hiểu sơ lược về kiểu dữ liệu chuỗi trong Javascript.

Ở bài này, Kteam xin hướng dẫn cho các bạn về **các phương thức đối với chuỗi trong Javascript**

Nội dung

Những nội dung các bạn cần nắm vững trước khi tìm hiểu bài này:

- Kiểu dữ liệu chuỗi trong Javascript

Và, ở bài này, chúng ta sẽ được tiếp cận về

- Các phương thức đối với chuỗi trong Javascript
 - Các phương thức xác định
 - Các phương thức tìm kiếm

Các phương thức đối với chuỗi trong Javascript

Các phương thức xác định

Phương thức at

Cú pháp:

```
<String>.at(<index>)
```

Tác dụng: Trả về ký tự nằm ở vị trí <index>.

Lưu ý: <index> có thể là giá trị âm, khi đó các phần tử được đếm ngược từ phải sang trái.

Lấy ví dụ với chuỗi "howKteam":

	h	o	w	K	t	e	a	m
Giá trị dương	0	1	2	3	4	5	6	7
Giá trị âm	-8	-7	-6	-5	-4	-3	-2	-1

Ví dụ về phương thức at:

Javascript:

```
a = "howKteam"
// 'howKteam'
a.at(3)
// 'K'
a.at(-4)
// 't'
a.at(-1)
// 'm'
a.at(10) // giá trị nằm ngoài phạm vi của chuỗi, trả về undefined
// undefined
a.at(-10) // giá trị nằm ngoài phạm vi của chuỗi, trả về undefined
// undefined
```

Phương thức includes

Cú pháp:

```
<String>.includes(<searchString>, <position>)
```

Tác dụng: Giúp xác định xem **<searchString>** có nằm trong **<String>** (từ **<position>** đến cuối chuỗi) hay không.

Các tham số trong phương thức:

- <searchString>**: là chuỗi được sử dụng để xác định, trả về **true** nếu được tìm thấy trong **<String>**, ngược lại trả về **false**.
- <position>**: Phương thức sẽ tìm kiếm từ vị trí **<position>** đến cuối chuỗi. Giá trị mặc định của nó là **0**.

Ví dụ:

Javascript:

```
a = "howKteam"
// 'howKteam'
a.includes("Kteam") // trong chuỗi a có "Kteam"
// true
a.includes("Kteam", 5) // từ a[5] đến cuối chuỗi, không xuất hiện "Kteam" (phạm vi tìm kiếm: "team")
// false
a.includes("how", 1)
// false
a.includes("howK")
// true
```

Phương thức startsWith

Cú pháp:

```
<String>.startsWith(<searchString>, <position>)
```

Tác dụng: Xác định xem **<String>** (vị trí từ **position** đến cuối chuỗi) có được bắt đầu bằng **<searchString>** hay không.

Trong đó:

- <searchString>**: là chuỗi được sử dụng để xác định, trả về **true** nếu chuỗi (từ **position** đến cuối chuỗi) được bắt đầu bằng **<searchString>**.
- <position>**: Là vị trí bắt đầu thực hiện xác định, mặc định bằng **0**.

Ví dụ:

Javascript:

```
a = 'howKteam'
// 'howKteam'
a.startsWith('how')
// true
a.startsWith('Kteam', 3) // từ vị trí thứ 3 đến cuối chuỗi: "Kteam", được bắt đầu bằng "Kteam"
// true
a.startsWith('Kt')
// false
a.startsWith('te', 4) // từ vị trí thứ 4 đến cuối chuỗi: "team", được bắt đầu bằng "te"
// true
```

Phương thức endsWith

Cú pháp:

```
<String>.endsWith(<searchString>, <length>)
```

Tác dụng: Xác định xem **<String>** có được kết thúc bằng **<searchString>** hay không.

Cách thức hoạt động gần giống với **startsWith**:

- **<searchString>**: là chuỗi được sử dụng để xác định, trả về **true** nếu chuỗi (bao gồm **length** kí tự đầu tiên của chuỗi) được kết thúc bằng **<searchString>**
- **<searchString>**: Là phạm vi xác định, cụ thể là lấy **<length>** ký tự đầu tiên của chuỗi. Giá trị mặc định là độ dài của chuỗi (**str.length**)

Ví dụ:

Javascript:

```
a = 'howKteam'
// 'howKteam'
a.endsWith("te", 6)
// true
a.endsWith("am")
// true
a.endsWith("eam")
// true
a.startsWith('ow')
// false
a.endsWith('ea')
// false
```

Các phương thức tìm kiếm

Phương thức indexOf

Cú pháp:

```
<String>.indexOf(<searchValue>, <fromIndex>)
```

Tác dụng: Tìm kiếm và trả về lần xuất hiện đầu tiên của **<searchValue>** trong **<String>**, phạm vi tìm kiếm bắt đầu từ **<fromIndex>**.

Các tham số:

- **<searchValue>**: Là chuỗi cần tìm kiếm.
- **<fromIndex>**: Là số nguyên, là giá trị bắt đầu cho việc tìm kiếm (hoạt động tìm kiếm được diễn ra từ **<fromIndex>** đến cuối **<String>**)

Giá trị trả về:

- Nếu tìm thấy, trả về vị trí đầu tiên mà **<searchValue>** được tìm thấy. Nếu không tìm thấy thì trả về **-1**
- Nếu **<searchValue>** không được đưa vào, hoặc **<fromIndex>** lớn hơn **<String>.length**, trả về **-1**

Ví dụ:

Javascript:

```
a = 'howKteam'
// 'howKteam'
a.indexOf("Kteam")
// 3
a.indexOf("Kteam", 3) // câu lệnh này và câu lệnh trên trả về cùng một giá trị, vì trong phạm vi tìm kiếm đều chứa "Kteam"
// 3
a.indexOf("Kteam", 4)
// -1
a.indexOf("Kteam", 100)
// -1
a.indexOf(100)
// -1
```

Phương thức lastIndexOf

Cú pháp:

<String>.lastIndexOf(<searchValue>, <fromIndex>)

Tác dụng: Gần giống như **indexOf**, nhưng lại tìm kiếm ở cuối chuỗi **<String>**. Cụ thể:

- **<searchValue>** giống như **indexOf**
- **<fromIndex>**: là một số nguyên, chỉ định việc tìm kiếm sẽ diễn ra trong đoạn **[0..<String>.length - fromIndex]** của **<String>**. Giá trị mặc định là **Infinity** (có thể hiểu, theo mặc định, thì việc tìm kiếm sẽ diễn ra trên toàn bộ chuỗi). Nếu **<fromIndex>** lớn hơn **<String>.length**, toàn bộ chuỗi được dùng để tìm kiếm

Ví dụ:

Javascript:

```
a = 'howKteam'
// 'howKteam'
a.lastIndexOf("Kt")
// 3
a.lastIndexOf("Kt", 3) // tìm kiếm trong đoạn [0..4] (7 - 3 = 4)
// 3
a.lastIndexOf("ho", 5)
// 0
```

Kết luận

Ở bài này, chúng ta đã tìm hiểu về các phương thức xác định + các phương thức tìm kiếm của chuỗi.

Bài tiếp theo, Kteam sẽ tiếp tục với chủ đề về [phương thức của chuỗi trong Javascript](#).

Cảm ơn các bạn đã theo dõi bài viết. Hãy để lại bình luận hoặc góp ý của mình để phát triển bài viết tốt hơn. Đừng quên **"Luyện tập – Thử thách – Không ngại khó"**.