MDN Web Docs
moz://a

String.prototype.slice()

The `slice()` method extracts a section of a string and returns it as a new string, without modifying the original string.

JavaScript Demo: String.slice()

```
const str = 'The quick brown fox jumps over the lazy dog.';

console.log(str.slice(31));
// expected output: "the lazy dog."

console.log(str.slice(4, 19));
// expected output: "quick brown fox"

console.log(str.slice(-4));
// expected output: "dog."

console.log(str.slice(-9, -5));
// expected output: "lazy"

```

Run ›

Reset

Syntax

```
slice(beginIndex)
slice(beginIndex, endIndex)
```

Parameters

`beginIndex`

The zero-based index at which to begin extraction.

If `beginIndex` is negative, `slice()` begins extraction from `str.length + beginIndex`. (E.g. `"test".slice(-2)` returns `"st"`)

If `beginIndex` is omitted, undefined, or cannot be converted to a number (using `Number(beginIndex)`), `slice()` begins extraction from the beginning of the string. (E.g. `"test".slice()` returns `"test"`)

If `beginIndex` is greater than or equal to `str.length`, an empty string is returned. (E.g. `"test".slice(4)` returns `""`)

`endIndex` **Optional**

The zero-based index *before* which to end extraction. The character at this index will not be included.

If `endIndex` is omitted, undefined, or cannot be converted to a number (using `Number(endIndex)`) `slice()` extracts to the end of the string. (E.g. `"test".slice(2)` returns `"st"`)

If `endIndex` is greater than `str.length`, `slice()` also extracts to the end of the string. (E.g. `"test".slice(2, 10)` returns `"st"`)

If `endIndex` is negative, `slice()` treats it as `str.length + endIndex`. (E.g, if `endIndex` is `-2`, it is treated as `str.length - 2` and `"test".slice(1, -2)` returns `"e"`).

If `endIndex` represents a position that is before the one represented by `startIndex`, `slice()` returns `""`. (E.g `"test".slice(2, -10)`, `"test".slice(-1, -2)` or `"test".slice(3, 2)`).

Return value

A new string containing the extracted section of the string.

Description

`slice()` extracts the text from one string and returns a new string. Changes to the text in one string do not affect the other string.

`slice()` extracts up to but not including `endIndex`. `str.slice(1, 4)` extracts the second

character through the fourth character (characters indexed `1`, `2`, and `3`).

As an example, `str.slice(2, -1)` extracts the third character through the second to last character in the string.

Examples

Using `slice()` to create a new string

The following example uses `slice()` to create a new string.

```
let str1 = 'The morning is upon us.', // the length of str1 is 23.
    str2 = str1.slice(1, 8),
    str3 = str1.slice(4, -2),
    str4 = str1.slice(12),
    str5 = str1.slice(30);
console.log(str2)  // OUTPUT: he morn
console.log(str3)  // OUTPUT: morning is upon u
console.log(str4)  // OUTPUT: is upon us.
console.log(str5)  // OUTPUT: ""
```

Using `slice()` with negative indexes

The following example uses `slice()` with negative indexes.

```
let str = 'The morning is upon us.'
str.slice(-3)      // returns 'us.'
str.slice(-3, -1)  // returns 'us'
str.slice(0, -1)   // returns 'The morning is upon us'
```

This example counts backwards from the end of the string by `11` to find the start index and forwards from the start of the string by `16` to find the end index.

```
console.log(str.slice(-11, 16)) // => "is u"
```

Here it counts forwards from the start by `11` to find the start index and backwards from the end by `7` to find the end index.

```
console.log(str.slice(11, -7)) // => " is u"
```

These arguments count backwards from the end by `5` to find the start index and backwards from the end by `1` to find the end index.

```
console.log(str.slice(-5, -1)) // => "n us"
```

Specifications

Specification
ECMAScript Language Specification (ECMAScript) # sec-string.prototype.slice

Browser compatibility

Report problems with this compatibility data on GitHub

slice	
Chrome	1
Edge	12
Firefox	1
Internet Explorer	4
Opera	4
Safari	1
WebView Android	1
Chrome Android	18
Firefox for Android	4
Opera Android	10.1
Safari on iOS	1
Samsung Internet	1.0
Deno	1.0
Node.js	0.10.0

Full support

See also

- `String.prototype.substr()`
- `String.prototype.substring()`
- `Array.prototype.slice()`

Last modified: Jan 6, 2022, by MDN contributors