JavaScript TUTORIAL

String.prototype.slice()

Summary: in this tutorial, you will learn how to use the JavaScript String `slice()` method to extract a substring from a string.

Introduction to the JavaScript String slice() method

The `String.prototype.slice()` method extracts a portion of a string and returns it as a substring. The following shows the syntax of the `slice()` method:

```
slice(start, end)
```

The `slice()` method has two optional parameters `start` and `end` .

start

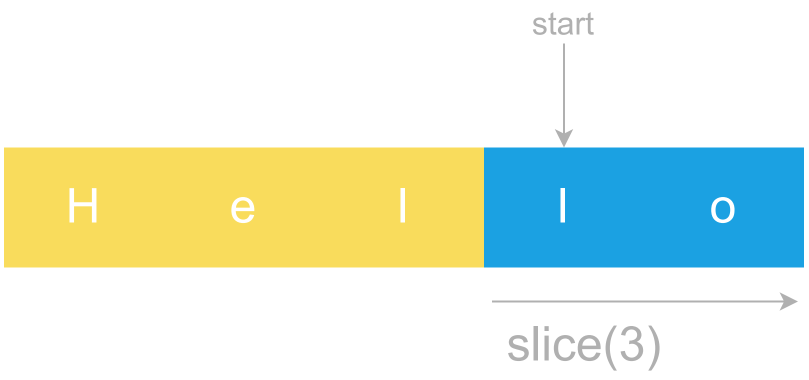
The `start` parameter is a zero-based index at which the method starts the extraction. For example:

```
const str = 'Hello';
const substr = str.slice(3);

console.log({ substr });
```

Output:

```
{ substr: 'lo' }
```



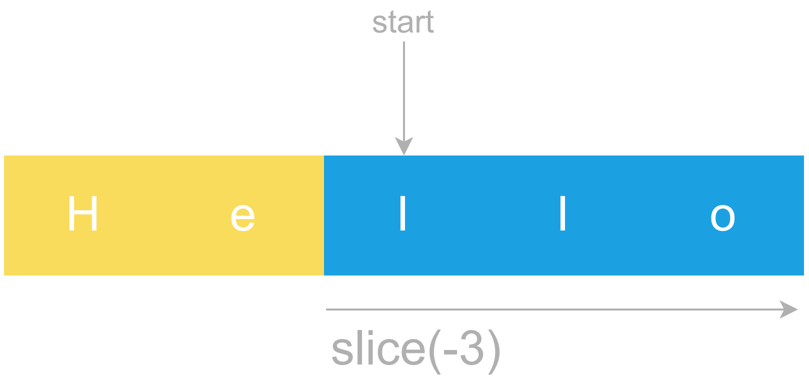
If the `start` is negative, the `slice()` method starts extraction from the `str.length + start`. For example:

```
const str = 'Hello';
const substr = str.slice(-3);

console.log({ substr });
```

Output:

```
{ substr: 'llo' }
```



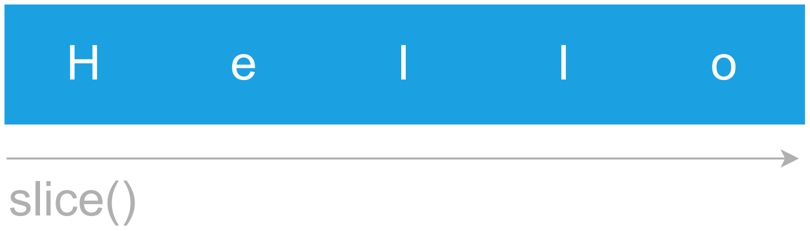
If the `start` is omitted, `undefined`, or cannot be converted to a number, the `slice()` method starts extraction from the beginning of the string:

```
const str = 'Hello';
const substr = str.slice();

console.log({ substr });
```

Output:

```
{ substr: 'Hello' }
```



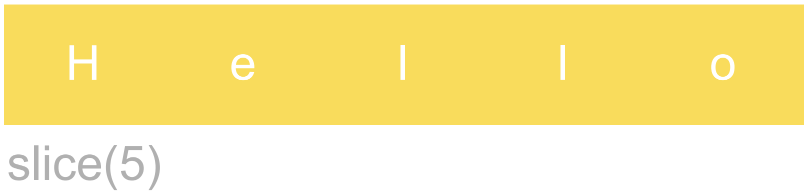
If the `start` is greater than or equal to the length of the string, the `slice()` method returns an empty string. For example:

```
const str = 'Hello';
const substr = str.slice(5);

console.log({ substr });
```

Output:

```
{ substr: '' }
```



end

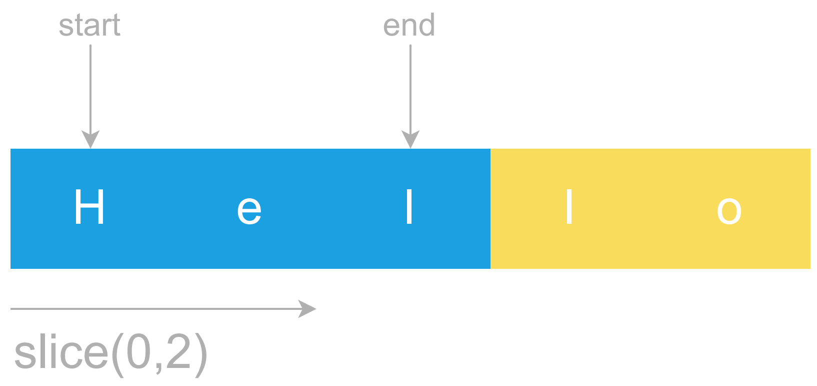
The `end` is a zero-based index that specifies the position before the `slice()` method ends the extraction. The result string will not include the character at the `end` index. For example:

```
const str = 'Hello';
const substr = str.slice(0, 2);
```

```
console.log({ substr });
```

Output:

```
{ substr: 'He' }
```



If the `end` is negative, the `slice()` method treats it as `str.length + end`. For example:

```
const str = 'Hello';
const substr = str.slice(0, -2);

// str.length 5
// str.length + end = 5 + (-2) = 3

console.log({ substr });
```

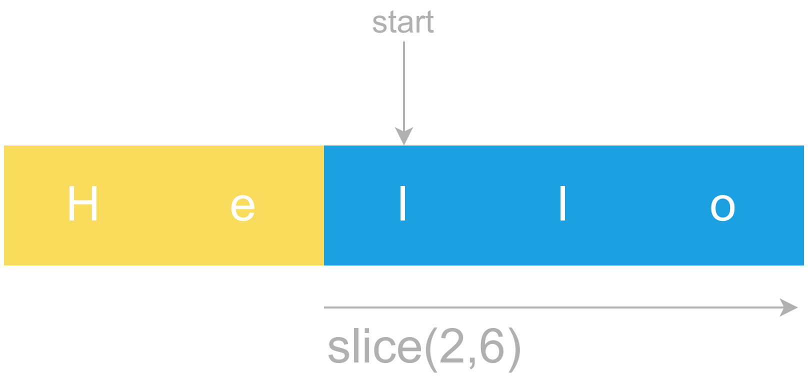
If the `end` is greater than the length of the string, the `slice()` method extracts to the end of the string. For example:

```
const str = 'Hello';
const substr = str.slice(2, 6);

console.log({ substr });
```

Output:

```
{ substr: 'llo' }
```



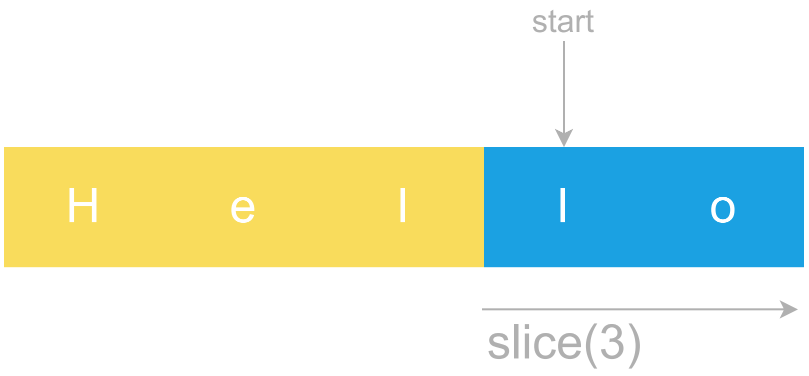
If the `end` is omitted, `undefined`, or cannot be converted to a number, the `slice()` method also extracts to the end of the string. For example:

```
const str = 'Hello';
const substr = str.slice(3);

console.log({ substr });
```

Output:

```
{ substr: 'lo' }
```



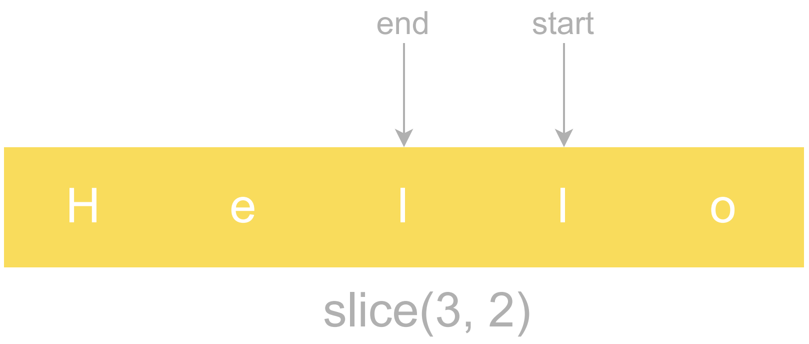
If the `end` represents a position that is before the one represented by the `start`, the `slice()` method returns an empty string. For example:

```
const str = 'Hello';
const substr = str.slice(3, 2);

console.log({ substr });
```

Output:

```
{ substr: '' }
```



JavaScript String slice() method practical example

The following example uses the `slice()` method to get the local part of an email address:

```
let email = 'john@example.com'
let localPart = email.slice(0,email.indexOf('@'));

console.log(localPart);
```

Output:

```
john
```

How it works:

- First, use the `indexOf()` method to locate the `@` sign. The returned value of the `indexOf()` is used as the second argument of the `slice()` method.

- Then, use the `slice()` method to extract the local part of the email starting from the beginning of the string up to the character before the `@` sign.

Summary

- Use the JavaScript String `slice()` method to extract a substring from a string.