

Related Topics

Стандартные встроенные объекты

String
Свойства
<code>String.length</code>
Методы
<code>String.prototype[@@iterator]()</code>
<div><div></div><code>String.prototype.anchor()</code></div>
<div><div></div><code>String.prototype.at()</code></div>
<div><div></div><code>String.prototype.big()</code></div>
<div><div></div><code>String.prototype.blink()</code></div>
<div><div></div><code>String.prototype.bold()</code></div>
<code>String.prototype.charAt()</code>
<code>String.prototype.charCodeAt()</code>
<code>String.prototype.codePointAt()</code>
<code>String.prototype.concat()</code>
<code>String.prototype.endsWith()</code>
<div><div></div><code>String.prototype.fixed()</code></div>
<div><div></div><code>String.prototype.fontcolor()</code></div>
<div><div></div><code>String.prototype.fontSize()</code></div>
<code>String.fromCharCode()</code>
<code>String.fromCharCode()</code>
<code>String.fromCodePoint()</code>
<code>String.prototype.includes()</code>
<code>String.prototype.indexOf()</code>
<div><div></div><code>String.prototype italics()</code></div>
<code>String.prototype.lastIndexOf()</code>
<div><div></div><code>String.prototype.link()</code></div>
<code>String.prototype.localeCompare()</code>
<code>String.prototype.match()</code>
<code>String.prototype.matchAll()</code>
<code>String.prototype.normalize()</code>
<code>String.prototype.padEnd()</code>
<code>String.prototype.padStart()</code>
<code>String.raw()</code>
<code>String.prototype.repeat()</code>
<code>String.prototype.replace()</code>
<code>String.prototype.replaceAll()</code>
<code>String.prototype.search()</code>
<code>String.prototype.slice()</code>
<div><div></div><code>String.prototype.small()</code></div>
<code>String.prototype.split()</code>
<code>String.prototype.startsWith()</code>
<div><div></div><code>String.prototype.strike()</code></div>
<div><div></div><code>String.prototype.sub()</code></div>
<div><div></div><code>String.prototype.substr()</code></div>
<code>String.prototype.substring()</code>
<div><div></div><code>String.prototype.sup()</code></div>
<code>String.prototype.toLocaleLowerCase()</code>
<code>String.prototype.toLocaleUpperCase()</code>
<code>String.prototype.toLowerCase()</code>
<div><div></div><div><div></div></div><code>String.prototype.toSource()</code></div>
<code>String.prototype.toString()</code>
<code>String.prototype.toUpperCase()</code>
<code>String.prototype.trim()</code>
<code>String.prototype.trimEnd()</code>
<code>String.prototype.trimStart()</code>
<code>String.prototype.valueOf()</code>

Наследование:
Function
Свойства
<div><div></div><code>Function.arguments</code></div>
<div><div></div><code>Function.caller</code></div>
<div><div></div><code>Function.displayName</code></div>
<code>Function.length</code>
<code>Function.name</code>
Методы
<code>Function.prototype.apply()</code>
<code>Function.prototype.bind()</code>
<code>Function.prototype.call()</code>
<div><div></div><div><div></div></div><code>Function.prototype.toSource()</code></div>
<code>Function.prototype.toString()</code>
Object
Свойства
<code>Object.prototype.constructor</code>
<div><div></div><code>Object.prototype.__proto__</code></div>
Методы
<div><div></div><code>Object.prototype.__defineGetter__()</code></div>
<div><div></div><code>Object.prototype.__defineSetter__()</code></div>
<div><div></div><code>Object.prototype.__lookupGetter__()</code></div>
<div><div></div><code>Object.prototype.__lookupSetter__()</code></div>
<code>Object.prototype.hasOwnProperty()</code>
<code>Object.prototype.isPrototypeOf()</code>
<code>Object.prototype.propertyIsEnumerable()</code>
<code>Object.setPrototypeOf()</code>
<code>Object.prototype.toLocaleString()</code>
<div><div></div><div><div></div></div><code>Object.prototype.toSource()</code></div>
<code>Object.prototype.toString()</code>
<code>Object.prototype.valueOf()</code>

String.prototype.substr()

[Report problems with this compatibility data on GitHub](#)

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

Full support

Deprecated. Not for use in new websites.

Внимание: так как `String.prototype.substr(...)` не полностью упрзднена (а лишь "удалена из стандартов по вебу"), она считается **унаследованной функцией**, использование которой лучше избегать там, где это возможно. Она не является частью ядра языка JavaScript и может быть удалена в будущем. Если есть возможность, используйте метод `substring()`.

Метод **substr()** возвращает указанное количество символов из строки, начиная с указанной позиции.

```
str.substr(start[, Length])
```

start

Позиция, с которой начинать извлекать символы. Если передано отрицательное число, она трактуется как `strLength - start`, где `strLength` равна длине строки (например, если параметр `start` равен -3, то он трактуется как `strLength - 3`.)

length

Необязательный параметр. Количество извлекаемых символов.

Новая строка, содержащая часть исходной, обрезанной по переданным параметрам.

Параметр `start` является индексом символа. Индекс первого символа равен 0, а последнего символа — на один меньше длины строки. Метод `substr()` начинает извлекать символы начиная с позиции `start` и собирая `length` символов (если он не достигнет раньше конца строки, в этом случае будет возвращено меньшее количество символов).

Если параметр `start` является положительным и он больше, либо равен длине строки, метод `substr()` вернёт пустую строку.

Если параметр `start` является отрицательным, метод `substr()` использует его как индекс символа, начиная с конца строки. Если параметр `start` отрицателен и по модулю больше длины строки, метод `substr()` будет использовать 0 в качестве начального индекса. Обратите внимание: описанная обработка отрицательных значений аргумента `start` не поддерживается JScript от Microsoft.

Если параметр `length` равен нулю или отрицателен, метод `substr()` вернёт пустую строку. Если параметр `length` опущен, метод `substr()` извлечёт все символы до конца строки.

```
var str = 'абвгдеёжи';

console.log('(1, 2): ' + str.substr(1, 2)); // '(1, 2): бв'
console.log('(-3, 2): ' + str.substr(-3, 2)); // '(-3, 2): жэ'
console.log('(-3): ' + str.substr(-3)); // '(-3): жи'
console.log('(1): ' + str.substr(1)); // '(1): бвгдеёжи'
console.log('(-20, 2): ' + str.substr(-20, 2)); // '(-20, 2): аб'
console.log('(20, 2): ' + str.substr(20, 2)); // '(20, 2): '
```

JScript от Microsoft не поддерживает отрицательные значения для начального индекса. Если вы хотите использовать эту возможность, вы можете использовать следующий код совместимости для обхода этой ошибки:

```
// only run when the substr() function is broken
if ('ab'.substr(-1) != 'b') {
  /**
   * Get the substring of a string
   * @param {integer} start where to start the substring
   * @param {integer} length how many characters to return
   * @return {string}
   */
  String.prototype.substr = function(substr) {
    return function(start, length) {
      // call the original method
      return substr.call(this,
        // did we get a negative start, calculate how much it is from the beginning of the string
        // adjust the start parameter for negative value
        start < 0 ? this.length + start : start,
        length)
    }
  }(String.prototype.substr);
}
```

Спецификация

[ECMAScript \(ECMA-262\)](#)
[Определение 'String.prototype.substr' в этой спецификации.](#)

- [String.prototype.slice\(\)](#)
- [String.prototype.substring\(\)](#)

Found a problem with this page?

- [Edit on GitHub](#)
- [Source on GitHub](#)
- [Report a problem with this content on GitHub](#)
- Want to fix the problem yourself? See [our Contribution guide](#).

Last modified: 20 окт. 2021 г., by [MDN contributors](#)

Change your language

Русский

Change language