# JavaScript Number Methods

## 1:The toString() Method

**toString()** returns a number as a string.

All number methods can be used on any type of numbers (literals, variables, or expressions):

2:The toExponential() Method

**toExponential()** returns a string, with a number rounded and written using exponential notation.

A parameter defines the number of characters behind the decimal point:

3:The toFixed() Method

**toFixed()** returns a string, with the number written with a specified number of decimals:

4:The toPrecision() Method

**toPrecision()** returns a string, with a number written with a specified length:

5:The valueOf() Method

**valueOf()** returns a number as a number.

# JavaScript String Methods

The **indexOf()** method returns the index of (the position of) the **first** occurrence of a specified text in a string:

The **lastIndexOf()** method returns the index of the **last** occurrence of a specified text in a string:

Both indexOf(), and lastIndexOf() return -1 if the text is not found.

Both methods accept a second parameter as the starting position for the search:

The **search()** method searches a string for a specified value and returns the position of the match:

The two methods, indexOf() and search(), are **equal?**

They accept the same arguments (parameters), and return the same value?

The two methods are **NOT** equal. These are the differences:

* The search() method cannot take a second start position argument.
* The indexOf() method cannot take powerful search values (regular expressions).

## Extracting String Parts

There are 3 methods for extracting a part of a string:

* slice(start, end)
* substring(start, end)
* substr(start, length)

## The slice() Method

**slice()** extracts a part of a string and returns the extracted part in a new string.

The method takes 2 parameters: the starting index (position), and the ending index (position).

Example:

var str = "Apple, Banana, Kiwi";

var res = str.slice(7, 13);

The result of res will be:

Banana

## The substring() Method

**substring()** is similar to slice().

The difference is that substring() cannot accept negative indexes.

### Example

var str = "Apple, Banana, Kiwi";

var res = str.substring(7, 13);

The result of *res* will be:

Banana

## The substr() Method

**substr()** is similar to slice().

The difference is that the second parameter specifies the **length** of the extracted part.

### Example

var str = "Apple, Banana, Kiwi";

var res = str.substr(7, 6);

The result of res will be:

Banana

## Replacing String Content

The **replace()** method replaces a specified value with another value in a string:

### Example

str = "Please visit Microsoft!";

var n = str.replace("Microsoft", "W3Schools");

## Converting to Upper and Lower Case

A string is converted to upper case with **toUpperCase()**:

### Example

var text1 = "Hello World!"; // String

var text2 = text1.toUpperCase(); // text2 is text1 converted to upper

A string is converted to lower case with **toLowerCase()**:

### Example

var text1 = "Hello World!"; // String

var text2 = text1.toLowerCase(); // text2 is text1 converted to lower

## The concat() Method

**concat()** joins two or more strings:

### Example

var text1 = "Hello";

var text2 = "World";

var text3 = text1.concat(" ", text2);

## String.trim()

String.trim() removes whitespace from both sides of a string.

### Example

var str = " Hello World! ";

alert(str.trim());

## Extracting String Characters

There are 2 **safe** methods for extracting string characters:

* charAt(position)
* charCodeAt(position)

## The charAt() Method

The **charAt()** method returns the character at a specified index (position) in a string:

### Example

var str = "HELLO WORLD";

str.charAt(0); // returns H

## The charCodeAt() Method

The **charCodeAt()** method returns the unicode of the character at a specified index in a string:

### Example

var str = "HELLO WORLD";

str.charCodeAt(0); // returns 72

# JavaScript Operators

## JavaScript Arithmetic Operators

Arithmetic operators are used to perform arithmetic on numbers:

|  |  |
| --- | --- |
| **Operator** | **Description** |
| + | Addition |
| - | Subtraction |
| \* | Multiplication |
| / | Division |
| % | Modulus (Division Remainder) |
| ++ | Increment |
| -- | Decrement |

## JavaScript Assignment Operators

Assignment operators assign values to JavaScript variables.

|  |  |  |
| --- | --- | --- |
| **Operator** | **Example** | **Same As** |
| = | x = y | x = y |
| += | x += y | x = x + y |
| -= | x -= y | x = x - y |
| \*= | x \*= y | x = x \* y |
| /= | x /= y | x = x / y |
| %= | x %= y | x = x % y |

## JavaScript Comparison Operators

|  |  |
| --- | --- |
| **Operator** | **Description** |
| == | equal to |
| === | equal value and equal type |
| != | not equal |
| !== | not equal value or not equal type |
| > | greater than |
| < | less than |
| >= | greater than or equal to |
| <= | less than or equal to |
| ? | ternary operator |