String object

String Object Properties

Property	Description
length	Returns the length of a string

String Object Methods

Method	Description
charAt()	Returns the character at the specified index
concat()	Joins two or more strings, and returns a copy of the joined strings
indexOf()	Returns the position of the first found occurrence of a specified value in a string
lastIndexOf()	Returns the position of the last found occurrence of a specified value in a string
match()	Searches for a match between a regular expression and a string, and returns the matches
replace()	Searches for a match between a substring (or regular expression) and a string, and replaces the matched substring with a new substring
search()	Searches for a match between a regular expression and a string, and returns the position of the match
slice()	Extracts a part of a string and returns a new string
split()	Splits a string into an array of substrings
substr()	Extracts the characters from a string, beginning at a specified start position, and through the specified number of character
substring()	Extracts the characters from a string, between two specified indices
toLowerCase()	Converts a string to lowercase letters
toUpperCase()	Converts a string to uppercase letters
valueOf()	Returns the primitive value of a String object

String HTML Wrapper Methods

The HTML wrapper methods return the string wrapped inside the appropriate HTML tag.

Method	Description
anchor()	Creates an anchor
big()	Displays a string using a big font
blink()	Displays a blinking string
bold()	Displays a string in bold
fixed()	Displays a string using a fixed-pitch font

fontcolor()	Displays a string using a specified color
fontsize()	Displays a string using a specified size
italics()	Displays a string in italic
link()	Displays a string as a hyperlink
small()	Displays a string using a small font
strike()	Displays a string with a strikethrough
sub()	Displays a string as subscript text
sup()	Displays a string as superscript tex

Properties

length

The String object is used to manipulate a stored piece of text.

Examples of use:

The following example uses the length property of the String object to find the length of a string:

```
var txt="Hello world!";
document.write(txt.length);
```

The code above will result in the following output:

```
12
```

Methods

Link () method Definition and Usage

The link() method is used to display a string as a hyperlink.

This method returns the string embedded in the <a> tag, like this:

```
<a href="url">string</a>
```

Syntax

string.link(url)

Parameter	Description
url	Required. The URL to link to

```
<script type="text/javascript">

var str = "Free Web Building Tutorials!";
document.write(str.link("http://www.w3schools.com"));

</script>
```

All Methods

```
<html>
<body>

<script type="text/javascript">

var txt = "Hello World!";

document.write("<p>Big: " + txt.big() + "</p>");
document.write("<p>Small: " + txt.small() + "</p>");

document.write("<p>Bold: " + txt.bold() + "</p>");
document.write("<p>Italic: " + txt.italics() + "</p>");

document.write("<p>Fixed: " + txt.fixed() + "</p>");
document.write("<p>Strike: " + txt.strike() + "</p>");

document.write("<p>Fontcolor: " + txt.fontcolor("green") + "</p>");
document.write("<p>Fontsize: " + txt.fontsize(6) + "</p>");

document.write("<p>Subscript: " + txt.sub() + "</p>");
document.write("<p>Superscript: " + txt.sup() + "</p>");

document.write("<p>Link: " + txt.link("http://www.w3schools.com") + "</p>");

document.write("<p>Blink: " + txt.blink() + " (does not work in IE, Chrome, or
Safari)</p>");

</script>

</body>
</html>
```

Output

Big: Hello World!

Small: Hello World!

Bold: **Hello World!**

Italic: *Hello World!*

Fixed: `Hello World!`

Strike: ~~Hello World!~~

Fontcolor: Hello World!

Fontsize: Hello World!

Subscript: Hello World!

Superscript: Hello World!

Link: Hello World!

Blink: Hello World! (does not work in IE, Chrome, or Safari)

replace() Definition and Usage

The replace() method searches for a match between a substring (or regular expression) and a string, and replaces the matched substring with a new substring

Syntax

string.replace(*regexp/substr*,*newstring*)

```
<html>
<body>

<script type="text/javascript">

var str="Visit Microsoft!";
document.write(str.replace("Microsoft","W3Schools"));

</script>
</body>
</html>
```

Output

```
Visit W3Schools!
```

substr() Definition and Usage

The substr() method extracts the characters from a string, beginning at "start" and through the specified number of character, and returns the new sub string.

Syntax

string.substr(start,length)

```
<html>
<body>

<script type="text/javascript">

var str="Hello world!";
document.write(str.substr(3)+"<br />");
document.write(str.substr(3,4));

</script>

</body>
</html>
```

Output

```
lo world!
lo w
```

substring() Definition and Usage

The substring() method extracts the characters from a string, between two specified indices, and returns the new sub string.

This method extracts the characters in a string between "from" and "to", not including "to" itself.

Syntax

string.substring(from, to)

```
<html>
<body>

<script type="text/javascript">

var str="Hello world!";
document.write(str.substring(3)+"<br />");
document.write(str.substring(3,7));

</script>

</body>
</html>
```

Output

```
lo world!
lo w
```

toLowerCase() Definition and Usage

The toLowerCase() method converts a string to lowercase letters.

Syntax

string.toLowerCase()

```
<html>
<body>

<script type="text/javascript">

var txt="Hello World!";
document.write(txt.toLowerCase() + "<br />");
document.write(txt.toUpperCase());

</script>

</body>
```

```
</html>
```

Output

```
hello world!
HELLO WORLD!
```

toUpperCase() Definition and Usage

The toUpperCase() method converts a string to uppercase letters.

Syntax

string.toUpperCase()

```
<html>
<body>

<script type="text/javascript">

var txt="Hello World!";
document.write(txt.toLowerCase() + "<br />");
document.write(txt.toUpperCase());

</script>

</body>
</html>
```

Output

```
hello world!
HELLO WORLD!
```

Math Object

Math Object Properties

Property	Description
E	Returns Euler's number (approx. 2.718)
LN2	Returns the natural logarithm of 2 (approx. 0.693)
LN10	Returns the natural logarithm of 10 (approx. 2.302)

LOG2E	Returns the base-2 logarithm of E (approx. 1.442)
LOG10E	Returns the base-10 logarithm of E (approx. 0.434)
PI	Returns PI (approx. 3.14159)
SQRT1_2	Returns the square root of 1/2 (approx. 0.707)
SQRT2	Returns the square root of 2 (approx. 1.414)

Definition and Usage

The E property returns the Euler's number and the base of natural logarithms, approximately 2.718.

Syntax

```
Math.E
```

Example:

```
<html>

<body>

<script type="text/javascript">

document.write("Euler's number: " + Math.E);

</script>

</body></html>
```

Output

```
Euler's number: 2.718281828459045
```

PI Definition and Usage

The PI property returns the ratio of a circle's area to the square of its radius, approximately 3.14159.

Syntax

Math.PI

```
<html>

<body>



<script type="text/javascript">

document.write("PI: " + Math.PI);

</script>

</body></html>
```

Output

```
PI: 3.141592653589793
```

Math Object Methods

Method	**Description**
abs(x)	Returns the absolute value of x
acos(x)	Returns the arccosine of x, in radians
asin(x)	Returns the arcsine of x, in radians
atan(x)	Returns the arctangent of x as a numeric value between -PI/2 and PI/2 radians
atan2(y,x)	Returns the arctangent of the quotient of its arguments
ceil(x)	Returns x, rounded upwards to the nearest integer
cos(x)	Returns the cosine of x (x is in radians)
exp(x)	Returns the value of E^x
floor(x)	Returns x, rounded downwards to the nearest integer
log(x)	Returns the natural logarithm (base E) of x
max(x,y,z,...,n)	Returns the number with the highest value
min(x,y,z,...,n)	Returns the number with the lowest value
pow(x,y)	Returns the value of x to the power of y
random()	Returns a random number between 0 and 1
round(x)	Rounds x to the nearest integer
sin(x)	Returns the sine of x (x is in radians)
sqrt(x)	Returns the square root of x
tan(x)	Returns the tangent of an angle

abs() Definition and Usage

The abs() method returns the absolute value of a number.

Syntax

Math.abs(x)

Parameter	Description
x	Required. A number

```
<html>
<body>

<script type="text/javascript">

document.write(Math.abs(7.25) + "<br />");
document.write(Math.abs(-7.25) + "<br />");
document.write(Math.abs(null) + "<br />");
document.write(Math.abs("Hello") + "<br />");
document.write(Math.abs(7.25-10));

</script>

</body>
</html>
```

Output

```
7.25
7.25
0
NaN
2.75
```

ceil() Definition and Usage

The ceil() method rounds a number UPWARDS to the nearest integer, and returns the result.

Syntax

```
Math.ceil(x)
```

```
<html>
<body>

<script type="text/javascript">

document.write(Math.ceil(0.60) + "<br />");
document.write(Math.ceil(0.40) + "<br />");
document.write(Math.ceil(5) + "<br />");
document.write(Math.ceil(5.1) + "<br />");
document.write(Math.ceil(-5.1) + "<br />");
document.write(Math.ceil(-5.9));

</script>

</body>
</html>
```

Output

```
1
1
5
6
-5
-5
```

floor() Definition and Usage

The floor() method rounds a number DOWNWARDS to the nearest integer, and returns the result.

Syntax

```
Math.floor(x)
```

```
<html>
<body>

<script type="text/javascript">
document.write(Math.floor(0.60) + "<br />");
document.write(Math.floor(0.40) + "<br />");
document.write(Math.floor(5) + "<br />");
document.write(Math.floor(5.1) + "<br />");
document.write(Math.floor(-5.1) + "<br />");
document.write(Math.floor(-5.9));
</script>

</body>
</html>
```

Output

```
0
0
5
5
-6
-6
```

exp() Definition and Usage

The exp() method returns the value of E^x, where E is Euler's number (approximately 2.7183) and x is the number passed to it.

Syntax

```
Math.exp(x)
```

```
<html>
<body>

<script type="text/javascript">
document.write(Math.exp(1) + "<br />");
document.write(Math.exp(-1) + "<br />");
document.write(Math.exp(5) + "<br />");
document.write(Math.exp(10) + "<br />");
</script>

</body>
</html>
```

Output

```
2.718281828459045
0.36787944117144233
```

```
148.4131591025766
22026.465794806718
```

cos() Definition and Usage

The cos() method returns the cosine of a number.

Note: The cos() method returns a numeric value between -1 and 1, which represents the cosine of the angle.

Syntax

```
Math.cos(x)
```

```
<html>
<body>

<script type="text/javascript">

document.write(Math.cos(3) + "<br />");
document.write(Math.cos(-3) + "<br />");
document.write(Math.cos(0) + "<br />");
document.write(Math.cos(Math.PI) + "<br />");
document.write(Math.cos(2*Math.PI));

</script>

</body>
</html>
```

Output

```
-0.9899924966004454
-0.9899924966004454
1
-1
1
```

sin() Definition and Usage

The sin() method returns the sine of a number.

Note: This method returns a numeric value between -1 and 1, which represents the sine of the parameter x.

Syntax

```
Math.sin(x)
```

```
<html>
<body>

<script type="text/javascript">

document.write(Math.sin(3) + "<br />");
document.write(Math.sin(-3) + "<br />");
document.write(Math.sin(0) + "<br />");
document.write(Math.sin(Math.PI) + "<br />");
document.write(Math.sin(Math.PI/2));

</script>

</body>
</html>
```

Output

```
0.1411200080598672
-0.1411200080598672
0
1.2246467991473532e-16
1
```

max() Definition and Usage

The max() method returns the number with the highest value.

Syntax

```
Math.max(x,y,z,...,n)
```

```
<html>
<body>
```

```
<script type="text/javascript">
document.write(Math.max(5,10) + "<br />");
document.write(Math.max(0,150,30,20,38) + "<br />");
document.write(Math.max(-5,10) + "<br />");
document.write(Math.max(-5,-10) + "<br />");
document.write(Math.max(1.5,2.5));
</script>

</body></html>
```

Output

```
10
150
10
-5
2.5
```

min() Definition and Usage

The min() method returns the number with the lowest value.

Syntax

```
Math.min(x,y,z,...,n)
```

```
<html>
<body>

<script type="text/javascript">

document.write(Math.min(5,10) + "<br />");
document.write(Math.min(0,150,30,20,38) + "<br />");
document.write(Math.min(-5,10) + "<br />");
document.write(Math.min(-5,-10) + "<br />");
document.write(Math.min(1.5,2.5));

</script>

</body>
</html>
```

Output

```
5
0
-5
-10
1.5
```

pow() Definition and Usage

The pow() method returns the value of x to the power of y (x^y).

Syntax

```
Math.pow(x,y)
```

```
<html>
<body>

<script type="text/javascript">

document.write(Math.pow(0,0) + "<br />");
document.write(Math.pow(0,1) + "<br />");
document.write(Math.pow(1,1) + "<br />");
document.write(Math.pow(1,10) + "<br />");
document.write(Math.pow(7,2) + "<br />");
document.write(Math.pow(-7,2) + "<br />");
document.write(Math.pow(2,4));

</script>

</body>
</html>
```

Output

```
1
0
1
1
49
49
16
```

random() Definition and Usage

The random() method returns a random number between 0 and 1.

Syntax

```
Math.random()
```

```
<html>
<body>

<script type="text/javascript">

//return a random number between 0 and 1
document.write(Math.random() + "<br />");

//return a random integer between 0 and 10
document.write(Math.floor(Math.random()*11));

</script>

</body>
</html>
```

Output

```
0.0998109087864345
10
```

round() Definition and Usage

The round() method rounds a number to the nearest integer.

Syntax

```
Math.round(x)
```

```
<html>
<body>

<script type="text/javascript">
```

```
document.write(Math.round(0.60) + "<br />");
document.write(Math.round(0.50) + "<br />");
document.write(Math.round(0.49) + "<br />");
document.write(Math.round(-4.40) + "<br />");
document.write(Math.round(-4.60));

</script>

</body>
</html>
```

Output

```
1
1
0
-4
-5
```

sin() Definition and Usage

The sin() method returns the sine of a number.

Note: This method returns a numeric value between -1 and 1, which represents the sine of the parameter x.

Syntax

```
Math.sin(x)
```

```
<html>
<body>

<script type="text/javascript">

document.write(Math.sin(3) + "<br />");
document.write(Math.sin(-3) + "<br />");
document.write(Math.sin(0) + "<br />");
document.write(Math.sin(Math.PI) + "<br />");
document.write(Math.sin(Math.PI/2));

</script>
```

```
</body>
</html>
```

Output

```
0.1411200080598672
-0.1411200080598672
0
1.2246467991473532e-16
1
```

sqrt() Definition and Usage

The sqrt() method returns the square root of a number.

Syntax

```
Math.sqrt(x)
```

```
<html>
<body>

<script type="text/javascript">

document.write(Math.sqrt(0) + "<br />");
document.write(Math.sqrt(1) + "<br />");
document.write(Math.sqrt(9) + "<br />");
document.write(Math.sqrt(0.64) + "<br />");
document.write(Math.sqrt(-9));

</script>

</body>
</html>
```

Output

```
0
1
3
0.8
NaN
```

Date Object Properties

Date Object Methods

Method	Description
getDate()	Returns the day of the month (from 1-31)
getDay()	Returns the day of the week (from 0-6)
getFullYear()	Returns the year (four digits)
getHours()	Returns the hour (from 0-23)
getMilliseconds()	Returns the milliseconds (from 0-999)
getMinutes()	Returns the minutes (from 0-59)
getMonth()	Returns the month (from 0-11)
getSeconds()	Returns the seconds (from 0-59)
getTime()	Returns the number of milliseconds since midnight Jan 1, 1970
getTimezoneOffset()	Returns the time difference between GMT and local time, in minutes
getUTCDate()	Returns the day of the month, according to universal time (from 1-31)
getUTCDay()	Returns the day of the week, according to universal time (from 0-6)
getUTCFullYear()	Returns the year, according to universal time (four digits)
getUTCHours()	Returns the hour, according to universal time (from 0-23)
getUTCMilliseconds()	Returns the milliseconds, according to universal time (from 0-999)
getUTCMinutes()	Returns the minutes, according to universal time (from 0-59)
getUTCMonth()	Returns the month, according to universal time (from 0-11)
getUTCSeconds()	Returns the seconds, according to universal time (from 0-59)
getYear()	Deprecated. Use the getFullYear() method instead
parse()	Parses a date string and returns the number of milliseconds since midnight of January 1, 1970
setDate()	Sets the day of the month (from 1-31)
setFullYear()	Sets the year (four digits)
setHours()	Sets the hour (from 0-23)
setMilliseconds()	Sets the milliseconds (from 0-999)

setMinutes()	Set the minutes (from 0-59)
setMonth()	Sets the month (from 0-11)
setSeconds()	Sets the seconds (from 0-59)
setTime()	Sets a date and time by adding or subtracting a specified number of milliseconds to/from midnight January 1, 1970

getDate() Definition and Usage

The getDate() method returns the day of the month (from 1 to 31) for the specified date, according to local time.

Note: This method is always used in conjunction with a Date object.

Syntax

Date.getDate()

```
<html>
<body>

<script type="text/javascript">
var d = new Date();
document.write(d.getDate());
</script>

</body>
</html>
```

Output

```
19
```

```
<html>
<body>

<script type="text/javascript">
var d=new Date("July 21, 1983 01:15:00");
document.write(d.getDate());
</script>

</body>
</html>
```

Output

```
21
```

getDay() Definition and Usage

The getDay() method returns the day of the week (from 0 to 6) for the specified date, according to local time.

Note: Sunday is 0, Monday is 1, and so on.

Note: This method is always used in conjunction with a Date object.

Syntax

Date.getDay()

```
<html>
<body>

<script type="text/javascript">
var d=new Date();
document.write(d.getDay());
</script>

</body>
</html>
```

Output

```
2
```

getFullYear() Definition and Usage

The getFullYear() method returns the year (four digits) of the specified date, according to local time.

Note: This method is always used in conjunction with a Date object.

Syntax

Date.getFullYear()

```
<html>
<body>

<script type="text/javascript">

var d=new Date();
document.write(d.getFullYear());

</script>

</body>
</html>
```

Output

```
2010
```

Definition and Usage

The getHours() method returns the hour (from 0 to 23) of the specified date and time, according to local time.

Note: This method is always used in conjunction with a Date object.

Syntax

Date.getHours()

```
<html>
<body>

<script type="text/javascript">
var d = new Date();
document.write(d.getHours());
</script>

</body>
</html>
```

Output

14

getMilliseconds() Definition and Usage

The getMilliseconds() method returns the milliseconds (from 0 to 999) of the specified date and time, according to local time.

Note: This method is always used in conjunction with a Date object.

Syntax

Date.getMilliseconds()

```
<html>
<body>

<script type="text/javascript">
var d = new Date();
document.write(d.getMilliseconds());
</script>

</body>
</html>
```

Output

280

getMinutes() Definition and Usage

The getMinutes() method returns the minutes (from 0 to 59) of the specified date and time, according to local time.

Note: This method is always used in conjunction with a Date object.

Syntax

Date.getMinutes()

Output

```
<html>
<body>

<script type="text/javascript">

var d = new Date();
document.write(d.getMinutes());

</script>

</body>
</html>
```

Output

```
27
```

getMonth() Definition and Usage

The getMonth() method returns the month (from 0 to 11) for the specified date, according to local time.

Note: January is 0, February is 1, and so on.

Note: This method is always used in conjunction with a Date object.

Syntax

Date.getMonth()

```
<html>
<body>

<script type="text/javascript">

var d = new Date();
document.write(d.getMonth());

</script>

</body>
</html>
```

Output

```
9
```

getSeconds() Definition and Usage

The getSeconds() method returns the seconds (from 0 to 59) of the specified date and time, according to local time.

Note: This method is always used in conjunction with a Date object.

Syntax

Date.getSeconds()

```
<html>
<body>

<script type="text/javascript">
var d=new Date();
document.write(d.getSeconds());
</script>

</body>
</html>
```

Output

```
56
```

getTime() Definition and Usage

The getTime() method returns the number of milliseconds since midnight of January 1, 1970 and the specified date.

Note: This method is always used in conjunction with a Date object.

Syntax

Date.getTime()

```
<html>
<body>

<script type="text/javascript">
var d=new Date();
document.write(d.getTime() + " milliseconds since 1970/01/01");
</script>

</body>
</html>
```

Output

```
1287478834330 milliseconds since 1970/01/01
```

setDate() Definition and Usage

The setDate() method sets the day of the month (from 1 to 31), according to local time.

Note: This method is always used in conjunction with a Date object.

Syntax

Date.setDate(day)

```
<html>
<body>

<script type="text/javascript">

var d = new Date();
d.setDate(15);
document.write(d);

</script>

</body>
</html>
```

Output

```
Fri Oct 15 2010 14:32:23 GMT+0530 (India Standard Time)
```

setFullYear() Definition and Usage

The setFullYear() method sets the year (four digits), according to local time.

Note: This method is always used in conjunction with a Date object.

Syntax

Date.setFullYear(year,month,day)

```
<html>
<body>

<script type="text/javascript">

var d = new Date();
d.setFullYear(2020);
document.write(d);

</script>

</body>
</html>
```

Output

```
Mon Oct 19 2020 14:33:08 GMT+0530 (India Standard Time)
```

setHours() Definition and Usage

The setHours() method sets the hour (from 0 to 23), according to local time.

Note: This method is always used in conjunction with a Date object.

Syntax

Date.setHours(hour,min,sec,millisec)

```
<html>
<body>
```

```
<script type="text/javascript">

var d = new Date();
d.setHours(15);
document.write(d);

</script>

</body>
</html>
```

Output

```
Tue Oct 19 2010 15:33:48 GMT+0530 (India Standard Time)
```

setMinutes() Definition and Usage

The setMinutes() method sets the minutes (from 0 to 59), according to local time.

Note: This method is always used in conjunction with a Date object.

Syntax

Date.setMinutes(min,sec,millisec)

```
<html>
<body>

<script type="text/javascript">

var d = new Date();
d.setMinutes(1);
document.write(d);

</script>

</body>
</html>
```

Output

```
Tue Oct 19 2010 14:01:41 GMT+0530 (India Standard Time)
```

setMonth() Definition and Usage

The setMonth() method sets the month (from 0 to 11), according to local time.

Note: January is 0, February is 1, and so on.

Note: This method is always used in conjunction with a Date object.

Syntax

Date.setMonth(month,day)

```
<html>
<body>

<script type="text/javascript">

var d=new Date();
d.setMonth(0);
document.write(d);

</script>

</body>
</html>
```

Output

```
Tue Jan 19 2010 14:37:19 GMT+0530 (India Standard Time)
```

setTime() Definition and Usage

The setTime() method sets a date and time by adding or subtracting a specified number of milliseconds to/from midnight January 1, 1970.

Note: This method is always used in conjunction with a Date object.

Syntax

Date.setTime(millisec)

```
<html>
<body>

<script type="text/javascript">

var d=new Date();
d.setTime(77771564221);
document.write(d);

</script>

</body>
</html>
```

Output

```
Mon Jun 19 1972 08:42:44 GMT+0530 (India Standard Time)
```

Array Object

The Array object is used to store multiple values in a single variable.

Array Object Properties

Property	Description
length	Sets or returns the number of elements in an array

length Definition and Usage

The length property sets or returns the number of elements in an array.

Syntax

array.length

```
<html>

<body>
```

```
<script type="text/javascript">


var fruits = ["Banana", "Orange", "Apple", "Mango"];

document.write("Original length: " + fruits.length);

document.write("<br />");

fruits.length=5;

document.write("New length: " + fruits.length);

</script></body></html>
```

Output

```
Original length: 4
New length: 5
```

Array Object Methods

Method	Description
concat()	Joins two or more arrays, and returns a copy of the joined arrays
join()	Joins all elements of an array into a string
pop()	Removes the last element of an array, and returns that element
push()	Adds new elements to the end of an array, and returns the new length
reverse()	Reverses the order of the elements in an array
shift()	Removes the first element of an array, and returns that element
slice()	Selects a part of an array, and returns the new array
sort()	Sorts the elements of an array
splice()	Adds/Removes elements from an array
toString()	Converts an array to a string, and returns the result
unshift()	Adds new elements to the beginning of an array, and returns the new length
valueOf()	Returns the primitive value of an array

concat() Definition and Usage

The concat() method is used to join two or more arrays.

This method does not change the existing arrays, it only returns a copy of the joined arrays.

Syntax

array.concat(array2, array3, ..., arrayX);

```
<html>

<body>

<script type="text/javascript">

var parents = ["Jani", "Tove"];

var children = ["Cecilie", "Lone"];

var family = parents.concat(children);

document.write(family);

</script>

</body>

</html>
```

Output

```
Jani,Tove,Cecilie,Lone
```

join() Definition and Usage

The join() method joins all elements of an array into a string, and returns the string.

The elements will be separated by a specified separator. The default separator is comma (,).

Syntax

array.join(separator)

```
<html>
```

```
<body>

<script type="text/javascript">

var fruits = ["Banana", "Orange", "Apple", "Mango"];

document.write(fruits.join() + "<br />");

document.write(fruits.join("+") + "<br />");

document.write(fruits.join(" and "));

</script>

</body></html>
```

Output

```
Banana,Orange,Apple,Mango
Banana+Orange+Apple+Mango
Banana and Orange and Apple and Mango
```

pop() Definition and Usage

The pop() method removes the last element of an array, and returns that element.

Note: This method changes the length of an array!

Syntax

array.pop()

```
<html>

<body>

<script type="text/javascript">

var fruits = ["Banana", "Orange", "Apple", "Mango"];

document.write(fruits.pop() + "<br />");

document.write(fruits + "<br />");

document.write(fruits.pop() + "<br />");
document.write(fruits);

</script>

</body></html>
```

Output

```
Mango
Banana,Orange,Apple
Apple
Banana,Orange
```

push() Definition and Usage

The push() method adds new elements to the end of an array, and returns the new length.

Note: This method changes the length of an array!

Syntax

array.push(element1, element2, ..., elementX)

```
<html>

<body>

<script type="text/javascript">

var fruits = ["Banana", "Orange", "Apple", "Mango"];

document.write(fruits.push("Kiwi") + "<br />");

document.write(fruits.push("Lemon","Pineapple") + "<br />");

document.write(fruits);

</script>

</body></html>
```

Output

```
5
7
Banana,Orange,Apple,Mango,Kiwi,Lemon,Pineapple
```

reverse() Definition and Usage

The reverse() method reverses the order of the elements in an array (makes the last element first, and the first element last).

Note: This method changes the original array!

Syntax

array.reverse()

```
<html>
<body>
<script type="text/javascript">
var fruits = ["Banana", "Orange", "Apple", "Mango"];
document.write(fruits.reverse());
</script>
</body></html>
```

Output

```
Mango,Apple,Orange,Banana
```

shift() Definition and Usage

The shift() method removes the first element of an array, and returns that element.

Note: This method changes the length of an array!

Syntax

array.shift()

```
<html>
<body>
<script type="text/javascript">
var fruits = ["Banana", "Orange", "Apple", "Mango"];
document.write(fruits.shift() + "<br />");
document.write(fruits + "<br />");
document.write(fruits.shift() + "<br />");
document.write(fruits);
</script>
</body></html>
```

```
Banana
Orange,Apple,Mango
Orange
Apple,Mango
```

sort() Definition and Usage

The sort() method sorts the elements of an array.

Note: This method changes the original array!

Syntax

array.sort(sortfunc)

```
<html>
<body>
<script type="text/javascript">
var fruits = ["Banana", "Orange", "Apple", "Mango"];
document.write(fruits.sort());
```

```
</script>

</body></html
```

Output

```
Apple,Banana,Mango,Orange
```