UNIT II

JavaScript Object: A javaScript object is an entity having state and behavior (properties and method). For example: car, pen, bike, chair, glass, keyboard, monitor etc.

JavaScript is an object-based language. Everything is an object in JavaScript.

JavaScript String Methods

Let's see the list of JavaScript string methods with examples.

Methods	Description
charAt()	It provides the char value present at the specified index.
concat()	It provides a combination of two or more strings.
indexOf()	It provides the position of a char value present in the given string.
search()	It searches a specified regular expression in a given string and returns its position if a match occurs.
replace()	It replaces a given string with the specified replacement.
substr()	It is used to fetch the part of the given string on the basis of the specified starting position and length.
substring()	It is used to fetch the part of the given string on the basis of the specified index.
toLowerCase()	It converts the given string into lowercase letter.

toLocaleLowerCase()	It converts the given string into lowercase letter on the basis of host?s current locale.
toUpperCase()	It converts the given string into uppercase letter.
toString()	It provides a string representing the particular object.
valueOf()	It provides the primitive value of string object.

Javascript charAt Method

```
<!DOCTYPE html>
<html>
<body>
<script>
var str="javascript";
document.write(str.charAt(2));
</script>
</body>
</html>
```

JavaScript String concat(str) Method

The JavaScript String concat(str) method concatenates or joins two strings.

```
<script>
var s1="javascript ";
var s2="concat";
var s3=s1.concat(s2);
document.write(s3);
</script>
```

JavaScript String toLowerCase() Method

The JavaScript String toLowerCase() method returns the given string in lowercase letters.

```
<!DOCTYPE html>
<html>
<body>
<script>
var s1="JavaScript toLowerCase Example";
var s2=s1.toLowerCase();
document.write(s2);
</script>
</body>
</html>
```

O/P: javascript tolowercase example

JavaScript String toUpperCase() Method

The JavaScript String toUpperCase() method returns the given string in uppercase letters.

```
<!DOCTYPE html>
<html>
<body>
<script>
var s1="JavaScript toUpperCase Example";
var s2=s1.toUpperCase();
document.write(s2);
```

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

O/P: JAVASCRIPT TOUPPERCASE EXAMPLE

JavaScript String replace() Method Example

Let's see some examples of replace() method.

```
<!DOCTYPE html>
<html>
<body>

<script>
var str="Javatpoint";
document.writeln(str.replace("tpoint","Script"));
</script>
</body>
</html>
```

o/p: JavaScript

JavaScript Date Object

The **JavaScript date** object can be used to get year, month and day. You can display a timer on the webpage by the help of JavaScript date object.

You can use different Date constructors to create date object. It provides methods to get and set day, month, year, hour, minute and seconds.

You can use 4 variant of Date constructor to create date object.

1. Date()

2. Date(milliseconds)
3. Date(dateString)
4. Date(year, month, day, hours, minutes, seconds, milliseconds)

Methods	Description
getDate()	It returns the integer value between 1 and 31 that represents the day for the specified date on the basis of local time.
getDay()	It returns the integer value between 0 and 6 that represents the day of the week on the basis of local time
getMilliseconds()	It returns the integer value between 0 and 999 that represents the milliseconds on the basis of local time.
getMinutes()	It returns the integer value between 0 and 59 that represents the minutes on the basis of local time.
getMonth()	It returns the integer value between 0 and 11 that represents the month on the basis of local time.

JavaScript Date Example

```
<html>
<body>

<script>
var today=new Date();
document.write(today);
</script>
</body>
</html>
```

o/p: Tue Oct 06 2020 17:11:29 GMT+0530 (India Standard Time)

JavaScript Math

The **JavaScript math** object provides several constants and methods to perform mathematical operation. Unlike date object, it doesn't have constructors.

JavaScript Math Methods

Let's see the list of JavaScript Math methods with description

Methods	Description
abs()	It returns the absolute value of the given number.
acos()	It returns the arccosine of the given number in radians.
asin()	It returns the arcsine of the given number in radians.
atan()	It returns the arc-tangent of the given number in radians.
cbrt()	It returns the cube root of the given number.
ceil()	It returns a smallest integer value, greater than or equal to the given num
cos()	It returns the cosine of the given number.
cosh()	It returns the hyperbolic cosine of the given number.
exp()	It returns the exponential form of the given number.
floor()	It returns largest integer value, lower than or equal to the given number.
hypot()	It returns square root of sum of the squares of given numbers.
log()	It returns natural logarithm of a number.

max()	It returns maximum value of the given numbers.
min()	It returns minimum value of the given numbers.
pow()	It returns value of base to the power of exponent.
random()	It returns random number between 0 (inclusive) and 1 (exclusive).
round()	It returns closest integer value of the given number.
sign()	It returns the sign of the given number
sin()	It returns the sine of the given number.
sinh()	It returns the hyperbolic sine of the given number.
sqrt()	It returns the square root of the given number
tan()	It returns the tangent of the given number.
tanh()	It returns the hyperbolic tangent of the given number.
trunc()	It returns an integer part of the given number.

Math.sqrt(n)

The JavaScript math.sqrt(n) method returns the square root of the given number.

```
<!DOCTYPE html>

<html>

<body>

Square Root of 16:

<script>

var x=Math.sqrt(16);
```

```
document.write(x);

</script>

</body>

</html>
```

o/p: Square Root is: 4

```
<!DOCTYPE html>

<html>

<body>

2 to the power of 5 is: <span id="p3"></span>

<script>

var L=Math.pow(2,5);

document.write(L);

</script>

</body>

</html>
```

o/p: 2 to the power of 5 is: 32

JavaScript Arrays:

What is an Array?

An array is a special variable, which can hold more than one value at a time.

```
<!DOCTYPE html>
<html>
<body>
```

```
<h2>JavaScript Arrays</h2>
<script>
var cars = ["Saab", "Volvo", "BMW"];
document.write(cars);
</script>
</body>
</html>
```

O/P:
JavaScript Arrays
Saab,Volvo,BMW

1)
```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Arrays</h2>
<script>
var cars = [
  "Saab",
  "Volvo",
  "BMW"
];
document.write(cars);
</script>

</body>
</html>
```

3) **Using the JavaScript Keyword new**

The following example also creates an Array, and assigns values to it:

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Arrays</h2>
<script>
var cars = new Array("Saab", "Volvo", "BMW");
document.write(cars);
</script>
```

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

Document Object Model:

The **document object** represents the whole html document

The DOM Programming Interface

The HTML DOM can be accessed with JavaScript (and with other programming languages).

In the DOM, all HTML elements are defined as **objects**.

The programming interface is the properties and methods of each object.

A **property** is a value that you can get or set (like changing the content of an HTML element).

A **method** is an action you can do (like add or deleting an HTML element).

Example

The following example changes the content (the innerHTML) of the <p> element with id="demo":

```
<!DOCTYPE html>
<html>
<body>

<h2>My First Page</h2>
<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = "Hello World!";
</script>

</body>

</html>
```

o/p: My First Page

Hello World!

In the example above, getElementById is a **method**, while innerHTML is a **property**.

The getElementById Method

The most common way to access an HTML element is to use the id of the element.

In the example above the getElementById method used id="demo" to find the element.

The innerHTML Property

The easiest way to get the content of an element is by using the innerHTML property.

The innerHTML property is useful for getting or replacing the content of HTML elements.

The innerHTML property can be used to get or change any HTML element, including <html> and <body>.

Ex:02

<!DOCTYPE html>

```
<html>
<body>

<h2>JavaScript String Methods</h2>

<p>The concat() method joins two or more strings:</p>

<p id="demo"></p>

<script>
var text1 = "Hello";
var text2 = "World!";
var text3 = text1.concat(" ",text2);
document.getElementById("demo").innerHTML = text3;
</script>

</body>
</html>
```

Ex :03

```
<!doctype html>
<html>
<head>
<script>
function add(){
var a,b,c;
a=Number(document.getElementById("first").value);
```

```
b=Number(document.getElementById("second").value);
c= a + b;
document.getElementById("answer").value= c;
}
</script>
</head>
<body>
Enter the First number : <input id="first">
Enter the Second number: <input id="second">
<button onclick="add()">Add</button>
<input id="answer">
</body>
</html>
```