**JavaScript**

**JavaScript** is a object-based scripting language. It is lightweight and cross-platform programming language.

It is not compiled but translated. JavaScript Translator (embedded in browser) is responsible to translate the JavaScript code.

JavaScript is used to create interactive websites. It is mainly used for:

1. Client-side validation
2. Dynamic drop-down menus
3. Displaying data and time
4. Displaying popup [windows[http://cdncache1-a.akamaihd.net/items/it/img/arrow-10x10.png](http://www.javatpoint.com/javascript-tutorial)](http://www.javatpoint.com/javascript-tutorial) and dialog boxes (like alert dialog box, confirm dialog box and prompt dialog box)
5. Displaying clocks etc.

# Understanding HTML/DOM events for Javascript

Javascript code is executed with events. So before learning javascript, let’s have some idea about events.

|  |  |
| --- | --- |
| **Events** | **Description** |
| onclick | occurs when element is clicked. |
| ondblclick | occurs when element is double-clicked. |
| onfocus | occurs when an element gets focus such as button, input, text area etc. |
| onblur | occurs when form looses the focus from an element. |
| onsubmit | occurs when form is submitted. |
| onmouseover | occurs when mouse is moved over an element. |
| onmouseout | occurs when mouse is moved out from an element (after moved over). |
| onmousedown | occurs when mouse button is pressed over an element. |
| onmouseup | occurs when mouse is released from an element (after mouse is pressed). |
| onload | occurs when document, object or frameset is loaded. |
| onunload | occurs when body or frameset is unloaded. |
| onscroll | occurs when document is scrolled. |
| onresized | occurs when document is resized. |
| onreset | occurs when form is reset. |
| onkeydown | occurs when key is being pressed. |
| onkeypress | occurs when user presses the key. |
| onkeyup | occurs when key is released. |

# Javascript Example

Javascript is easy to code. Javascript provides you 3 places to have the code: wihin body tag, within head tag and external javascript file.

Let’s create the first javascript example.

1. <html>
2. <body>
3. <script type="text/javascript">
4. document.write("JavaScript is a simple language for javatpoint learners");
5. </script>
6. </body>
7. </html>

The **script** tag specifies that we are using javascript.

The **text/javascript** is the content type that provides information to the browser about the data.

The **document.write()** function is used to display dynamic content through javascript.

[JavaScript Introduction](http://www.javatpoint.com/javascript-tutorial)

[What is JavaScript](http://www.javatpoint.com/javascript-tutorial) [Understanding Events](http://www.javatpoint.com/understanding-html-dom-events) [JavaScript Example](http://www.javatpoint.com/javascript-example) [External JavaScript](http://www.javatpoint.com/external-javascript-file)

[Basic Elements](http://www.javatpoint.com/javascript-comment)

[Comment](http://www.javatpoint.com/javascript-comment) [Variable](http://www.javatpoint.com/javascript-variable) [Global Variable](http://www.javatpoint.com/javascript-global-variable) [Data Types](http://www.javatpoint.com/javascript-data-types) [operators](http://www.javatpoint.com/javascript-operators) [If Statement](http://www.javatpoint.com/javascript-if) [Switch](http://www.javatpoint.com/javascript-switch) [Loop: for and while](http://www.javatpoint.com/javascript-loop) [Function](http://www.javatpoint.com/javascript-function)

[JavaScript Objects](http://www.javatpoint.com/javascript-objects)

[JavaScript Object](http://www.javatpoint.com/javascript-objects) [JavaScript Array](http://www.javatpoint.com/javascript-array)

[Browser Object Model](http://www.javatpoint.com/browser-object-model)

[Browser Objects](http://www.javatpoint.com/browser-object-model) [1) Window Object](http://www.javatpoint.com/window-object)

[2) Document Object](http://www.javatpoint.com/document-object-model) [getElementById](http://www.javatpoint.com/document-getElementById%28%29-method) [getElementsByName](http://www.javatpoint.com/document-getElementsByName%28%29-method) [getElementsByTagName](http://www.javatpoint.com/document-getElementsByTagName%28%29-method) [innerHTML property](http://www.javatpoint.com/javascript-innerHTML) [innerText property](http://www.javatpoint.com/javascript-innerText)

[JavaScript Validation](http://www.javatpoint.com/javascript-form-validation)

[form validation](http://www.javatpoint.com/javascript-form-validation) [email validation](http://www.javatpoint.com/javascript-form-validation#email)

|  |
| --- |
|  |
|  |
| 1) Javascript code between the body tag of html | |

1. <html>
2. <body>
3. <script type="text/javascript">
4. alert("Hello Javatpoint");
5. </script>
6. </body>
7. </html>

**2)** Javascript code between the head tag of html

we are creating a function msg(). To create function in javascript, you need to write function with functionname as given below.

To call function, you need to work on event. Here we are using onclick event to call msg() function.

1. <html>
2. <head>
3. <script type="text/javascript">
4. function msg(){
5. alert("Hello Javatpoint");
6. }
7. </script>
8. </head>
9. <body>
10. <p>Welcome to Javascript</p>
11. <form>
12. <input type="button" value="click" onclick="msg()"/>
13. </form>
14. </body>
15. </html>

# External JavaScript file

We can create the external JavaScript file and embed it in many html page. It provides code reusability because single JavaScript file can be used in several html pages

**message.js**

1. function msg(){
2. alert("Hello Javatpoint");
3. }

**index.html**

1. <html>
2. <head>
3. <script type="text/javascript" src="message.js"></script>
4. </head>
5. <body>
6. <p>Welcome to JavaScript</p>
7. <form>
8. <input type="button" value="click" onclick="msg()"/>
9. </form>
10. </body>
11. </html>

# Javascript Comment

The **javascript comments** are meaningful way to deliver message. It is used to add information about the code, warnings or suggestions so that the end user can easily interpret the code.

The javascript comment is ignored by the javascript engine embedded in the browser.

**Advantages of javascript comments**

There are mainly two advantages of javascript comments.

1. **Easy to understand** It can be used to elaborate the code so that it can be easy to understand.
2. **To avoid the unnecessary code** It can also be used to avoid the code being executed. Sometimes, we add the code to perform some action. But after sometime, there may be need to disable the code. In such case, it is better to use comments.

## Types of Javascript Comments

There are two types of comments in javascipt.

1. Single-line Comment
2. Multi-line Comment

### Single-line Comment

It is represented by double forward slashes (//). It can be used before and after the statement.

1. <script>
2. // It is single line comment
3. document.write("hello javascript");
4. </script>

Let’s see the example of single-line comment i.e. added after the statement.

1. <script>
2. var a=10;
3. var b=20;
4. var c=a+b;//It adds values of a and b variable
5. document.write(c);//prints sum of 10 and 20
6. </script>

### Multi-line Comment

It can be used to add single as well as multi line comments. So, it is more convenient.

It is represented by forward slash with asterisk then asterisk with forward slash. For example:

1. /\* your code here  \*/

It can be used before, after and middle of the statement.

1. <script>
2. /\* It is multi line comment.
3. It will not be displayed \*/
4. document.write("example of javascript multiline comment");
5. </script>

# Javascript Variable

A **variable in javascript** is simply a name of storage location. There are two types of variables in javascript : local and global.

There are some rules while declaring a variable (also known as identifiers).

1. Name must start with a letter (a to z or A to Z), underscore( \_ ), or dollar( $ ) sign.
2. After first letter we can use digits (0 to 9), for example value1.
3. Javascript variables are case sensitive, for example x and X are different variables.

## Example of correct javascript variables

1. var x = 10;
2. var \_value="sonoo";

## Example of incorrect javascript variables

1. var  123=30;
2. var \*aa=320;

#### Full example of javascript variable

1. <script>
2. var x = 10;
3. var y = 20;
4. var z=x+y;
5. document.write(z);
6. </script>

#### Output of the above example

30

## javascript local variable

A variable which is declared inside block or function is called local variable. It is accessible within the function or block only. For example:

1. <script>
2. function abc(){
3. var x=10;//local variable
4. }
5. </script>

Or,

1. <script>
2. If(10<13){
3. var y=20;//javascript local variable
4. }
5. </script>

## javascript global variable

A **global variable** is accessible from any function. A variable i.e. declared outside the function or declared with window object is known as global variable. For example:

 <script>

 var data=200;//gloabal variable

 function a(){

 document.write(data);

 }

 function b(){

 document.write(data);

 }



 </script>

# Javascript Data Types

Javascript provides different **data types** to hold different types of values. There are two types of data types in javascript.

1. Primitive data type
2. Non-primitive (reference) data type

Javascript is a **dynamic type language**, means you don't need to specify type of the variable because it is dynamically used by javascript engine. You need to use **var** here to specify the data type. It can hold any type of values such as numbers, strings etc. For example:

1. var a=40;//holding number
2. var b="Rahul";//holding string

There are five types of primitive data types in javascript. They are as follows:

1. String
2. Number
3. Boolean
4. Undefined
5. Null

The non-primitive data types are as follows:

* Object
* Array
* RegExp
* etc.

# JavaScript Operators

Operator is a symbol only. JavaScript operators are used to perform operations on operands. For example:

1. var sum=10+20;

Here, + is the arithmetic operator and = is the assignment operator.

There are following types of operators in JavaScript.

1. Arithmetic Operators
2. Comparison (Relational) Operators
3. Bitwise Operators
4. Logical Operators
5. Assignment Operators
6. Special Operators

### Arithmetic Operators

Arithmetic operators are used to perform arithmetic operations on the operands. The following operators are known as JavaScript arithmetic operators.

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| + | Addition | 10+20 = 30 |
| - | Subtraction | 20-10 = 10 |
| \* | Multiplication | 10\*20 = 200 |
| / | Division | 20/10 = 2 |
| % | Modulus (Remainder) | 20%10 = 0 |
| ++ | Increment | var a=10; a++; Now a = 11 |
| -- | Decrement | var a=10; a--; Now a = 9 |

### Comparison Operators

The JavaScript comparison operator compares the two operands. The comparison operators are as follows:

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| == | Is equal to | 10==20 = false |
| === | Identical (equal and of same type) | 10==20 = false |
| != | Not equal to | 10!=20 = true |
| !== | Not Identical | 20!==20 = false |
| > | Greater than | 20>10 = true |
| >= | Greater than or equal to | 20>=10 = true |
| < | Less than | 20<10 = false |
| <= | Less than or equal to | 20<=10 = false |

### Bitwise Operators

The bitwise operators perform bitwise operations on operands. The bitwise operators are as follows:

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| & | Bitwise AND | (10==20 & 20==33) = false |
| | | Bitwise OR | (10==20 | 20==33) = false |
| ^ | Bitwise XOR | (10==20 ^ 20==33) = false |
| ~ | Bitwise NOT | (~10) = -10 |
| << | Bitwise Left Shift | (10<<2) = 40 |
| >> | Bitwise Right Shift | (10>>2) = 2 |
| >>> | Bitwise Right Shift with Zero | (10>>>2) = 2 |

### Logical Operators

The following operators are known as JavaScript logical operators.

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| && | Logical AND | (10==20 && 20==33) = false |
| || | Logical OR | (10==20 || 20==33) = false |
| ! | Logical Not | !(10==20) = true |

### Assignment Operators

The following operators are known as JavaScript assignment operators.

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| = | Assign | 10+10 = 20 |
| += | Add and assign | var a=10; a+=20; Now a = 30 |
| -= | Subtract and assign | var a=20; a+=10; Now a = 10 |
| \*= | Multiply and assign | var a=10; a\*=20; Now a = 200 |
| /= | Divide and assign | var a=10; a/=2; Now a = 5 |
| %= | Modulus and assign | var a=10; a%=2; Now a = 0 |

### Special Operators

The following operators are known as JavaScript special operators.

|  |  |
| --- | --- |
| **Operator** | **Description** |
| (?:) | Conditional Operator returns value based on the condition. It is like if-else. |
| , | Comma Operator allows multiple expressions to be evaluated as single statement. |
| delete | Delete Operator deletes a property from the object. |
| in | In Operator checks if object has the given property |
| instanceof | checks if the object is an instance of given type |
| new | creates an instance (object) |
| typeof | checks the type of object. |
| void | it discards the expression's return value. |
| yield | checks what is returned in a generator by the generator's iterator. |

# JavaScript Switch

The **switch statement** is used in JavaScript to execute one code from multiple expressions. It is just like else if statement. But it is convenient than if..else..if because it can be used with numbers, characters etc.

Let’s see the simple example of switch statement in javascript.

1. <script>
2. var grade='B';
3. var result;
4. switch(grade){
5. case 'A':
6. result="A Grade";
7. break;
8. case 'B':
9. result="B Grade";
10. break;
11. case 'C':
12. result="C Grade";
13. break;
14. default:
15. result="No Grade";
16. }
17. document.write(result);
18. </script>

#### The switch statement is fall-through i.e. all the cases will be evaluated if you don't use break statement.

# JavaScript Loop

The **loops** are used to iterate the piece of code. They makes the code compact. It is mostly used in array.

There are four types of loops in JavaScript.

1. for loop
2. while loop
3. do-while loop
4. for-in loop

## For loop

The JavaScript for loop iterates the elements for the fixed number of times mostly. The syntax of for loop is given below.

1. for (initialization; condition; increment)
2. {
3. code to be executed
4. }

Let’s see the simple example of for loop in javascript.

1. <script>
2. for (i=1; i<=5; i++)
3. {
4. document.write(i + "<br/>")
5. }
6. </script>

## while loop

The syntax of while loop is given below.

1. while (condition)
2. {
3. code to be executed
4. }

Let’s see the simple example of while loop in javascript.

1. <script>
2. var i=11;
3. while (i<=15)
4. {
5. document.write(i + "<br/>");
6. i++;
7. }
8. </script>

#### Output of the above example

11  
12  
13  
14  
15

## do while loop

code is executed once always whether condition is true or false. The syntax of do while loop is given below.

1. do{
2. code to be executed
3. }while (condition);

Let’s see the simple example of do while loop in javascript.

1. <script>
2. var i=21;
3. do{
4. document.write(i + "<br/>");
5. i++;
6. }while (i<=25);
7. </script>

#### Output of the above example

21  
22  
23  
24  
25

## for in loop

The for in loop is used to iterate the properties of an object.

# JavaScript Function

The **functions** in JavaScript are used to perform operations. We can call function many times to reuse the code.

#### Advantage of function

There are mainly two advantages of functions.

1. **Code Reusability**: We can call a function several times so it save coding.
2. **Less Coding**: It makes our [program[http://cdncache1-a.akamaihd.net/items/it/img/arrow-10x10.png](http://www.javatpoint.com/javascript-function)](http://www.javatpoint.com/javascript-function) compact. We don’t need to write many lines of code each time to perform a common task.

The syntax of declaring function is given below.

1. function functionName([arg1, arg2, ...argN]){
2. //code to be executed
3. }

Function may have arguments or not.

Let’s see the simple example of function in javascript that does not has arguments.

1. <script>
2. function msg(){
3. alert("hello! this is message");
4. }
5. </script>
6. <input type="button" onclick="msg()" value="call function"/>

## Function Arguments

We can call function by passing arguments. Let’s see the example of function that has one argument.

1. <script>
2. function getcube(number){
3. alert(number\*number\*number);
4. }
5. </script>
6. <form>
7. <input type="button" value="click" onclick="getcube(4)"/>
8. </form>

## Function with Return Value

We can call function that returns a value and use it in our [program[http://cdncache1-a.akamaihd.net/items/it/img/arrow-10x10.png](http://www.javatpoint.com/javascript-function)](http://www.javatpoint.com/javascript-function). Let’s see the example of function that returns value.

1. <script>
2. function getInfo(){
3. return "hello javatpoint! How r u?";
4. }
5. </script>
6. <script>
7. document.write(getInfo());
8. </script>

# JavaScript Objects

An 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 is template based not class based. Here, we don't create class to get the object. But, we direct create objects.

## Creating Objects in JavaScript

There are 3 ways to create objects.

1. By object literal
2. By creating instance of Object directly (using new keyword)
3. By using an object constructor (using new keyword)

## 1) Creating Object by object literal

The syntax of creating object using object literal is given below:

1. object={property1:value1,property2:value2.....propertyN:valueN}

As you can see, property and value is separated by : ([colon[http://cdncache1-a.akamaihd.net/items/it/img/arrow-10x10.png](http://www.javatpoint.com/javascript-objects)](http://www.javatpoint.com/javascript-objects)).

Let’s see the simple example of creating object in JavaScript.

1. <script>
2. emp={id:102,name:"Shyam Kumar",salary:40000}
3. document.write(emp.id+" "+emp.name+" "+emp.salary);
4. </script>

#### Output of the above example

102 Shyam Kumar 40000

## 2) By creating instance of Object directly

The syntax of creating object directly is given below:

1. var objectname=new Object();

Here, **new keyword** is used to create object.

Let’s see the example of creating object directly.

1. <script>
2. var emp=new Object();
3. emp.id=101;
4. emp.name="Ravi Malik";
5. emp.salary=50000;
6. document.write(emp.id+" "+emp.name+" "+emp.salary);
7. </script>
8. </pre></div>
9. <h4 class="h4">Output of the above example</h4>
10. <div class="codeblock3">
11. <script>
12. var emp=new Object();
13. emp.id=101;
14. emp.name="Ravi";
15. emp.salary=50000;
16. document.write(emp.id+" "+emp.name+" "+emp.salary);
17. </script>

|  |
| --- |
|  |
|  |
| An 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 is template based not class based. Here, we don't create class to get the object. But, we direct create objects. Creating Objects in JavaScript There are 3 ways to create objects.   1. By object literal 2. By creating instance of Object directly (using new keyword) 3. By using an object constructor (using new keyword)  1) Creating Object by object literal The syntax of creating object using object literal is given below:   1. object={property1:value1,property2:value2.....propertyN:valueN}   As you can see, property and value is separated by : ([colon[http://cdncache1-a.akamaihd.net/items/it/img/arrow-10x10.png](http://www.javatpoint.com/javascript-objects)](http://www.javatpoint.com/javascript-objects)).  Let’s see the simple example of creating object in JavaScript.   1. <script> 2. emp={id:102,name:"Shyam Kumar",salary:40000} 3. document.write(emp.id+" "+emp.name+" "+emp.salary); 4. </script>  Output of the above example 102 Shyam Kumar 40000 2) By creating instance of Object directly The syntax of creating object directly is given below:   1. var objectname=new Object();   Here, **new keyword** is used to create object.  Let’s see the example of creating object directly.   1. <script> 2. var emp=new Object(); 3. emp.id=101; 4. emp.name="Ravi Malik"; 5. emp.salary=50000; 6. document.write(emp.id+" "+emp.name+" "+emp.salary); 7. </script> 8. </pre></div> 9. <h4 class="h4">Output of the above example</h4> 10. <div class="codeblock3"> 11. <script> 12. var emp=new Object(); 13. emp.id=101; 14. emp.name="Ravi"; 15. emp.salary=50000; 16. document.write(emp.id+" "+emp.name+" "+emp.salary); 17. </script>  3) By using an object constructor Here, you need to create function with arguments. Each argument value can be assigned in the current object by using this keyword.  The **this keyword** refers to the current object.  The example of creating object by object constructor is given below.   1. <script> 2. function emp(id,name,salary){ 3. this.id=id; 4. this.name=name; 5. this.salary=salary; 6. } 7. e=new emp(103,"Vimal Jaiswal",30000); 9. document.write(e.id+" "+e.name+" "+e.salary); 10. </script> | |

#### Output of the above example

103 Vimal Jaiswal 30000

## Defining method in JavaScript Object

We can define method in JavaScript object. But before defining method, we need to add property in the function with same name as method.

The example of defining method in object is given below.

1. <script>
2. function emp(id,name,salary){
3. this.id=id;
4. this.name=name;
5. this.salary=salary;
7. this.changeSalary=changeSalary;
8. function changeSalary(otherSalary){
9. this.salary=otherSalary;
10. }
11. }
12. e=new emp(103,"Sonoo Jaiswal",30000);
13. document.write(e.id+" "+e.name+" "+e.salary);
14. e.changeSalary(45000);
15. document.write("<br>"+e.id+" "+e.name+" "+e.salary);
16. </script>

#### Output of the above example

103 Sonoo Jaiswal 30000  
103 Sonoo Jaiswal 45000

# JavaScript Array

In JavaScript, **array** is an object that represents a collection of similar types of elements.

There are 3 ways to construct array in JavaScript

1. By array literal
2. By creating instance of Array directly (using new keyword)
3. By using an Array constructor (using new keyword)

## 1) By array literal

1. var arrayname=[value1,value2.....valueN];
2. </textarea></div>
3. <p>As you can see, values are contained inside [ ] and separated by , (comma).</p>
4. <p><p>Let’s see the simple example of creating and using array in JavaScript.</p></p>
5. <div class="codeblock"><textarea  name="code" class="xml">
6. <script>
7. var emp=["Sonoo","Vimal","Ratan"];
8. for (i=0;i<emp.length;i++){
9. document.write(emp[i] + "<br/>");
10. }
11. </script>

The .length property returns the length of an array.

**Output of the above example**

Sonoo  
Vimal  
Ratan

## 2) By creating instance of Array directly (using new keyword)

The syntax of creating array directly is given below:

1. var arrayname=new Array();

Here, **new keyword** is used to create instance of array.

Let’s see the example of creating array directly.

1. <script>
2. var i;
3. var emp = new Array();
4. emp[0] = "Arun";
5. emp[1] = "Varun";
6. emp[2] = "John";
8. for (i=0;i<emp.length;i++){
9. document.write(emp[i] + "<br>");
10. }
11. </script>

#### Output of the above example

Arun  
Varun  
John

## 3) By using an Array constructor (using new keyword)

Here, you need to create instance of array by passing arguments in constructor so that we don't have to provide value explicitely.

The example of creating object by array constructor is given below.

1. <script>
2. var emp=new Array("Jai","Vijay","Smith");
3. for (i=0;i<emp.length;i++){
4. document.write(emp[i] + "<br>");
5. }
6. </script>

#### Output of the above example

Jai  
Vijay  
Smith

# Browser Object Model

The **Browser Object Model** (BOM) is used to interact with the browser.

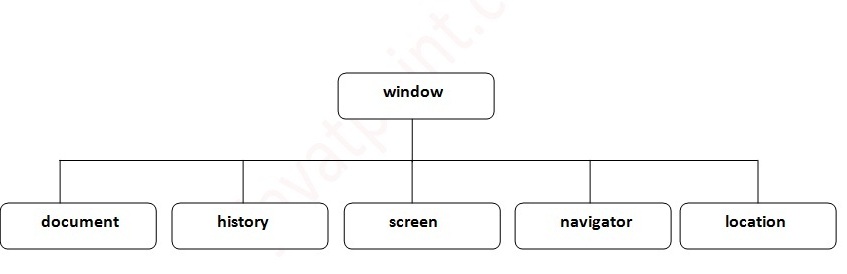
The default object of browser is window means you can call all the functions of window by specifying window or directly. For example:

1. window.alert("hello javatpoint");

is same as:

1. alert("hello javatpoint");
2. You can use a lot of properties (other objects) defined underneath the window object like document, [history[http://cdncache1-a.akamaihd.net/items/it/img/arrow-10x10.png](http://www.javatpoint.com/browser-object-model)](http://www.javatpoint.com/browser-object-model), screen, navigator, location, innerHeight, innerWidth,

#### Note: The document object represents an html document. It forms DOM (Document Object Model).



# Window Object

The **window object** represents a window in browser. An object of window is created automatically by the browser.

Window is the object of browser, **it is not the object of javascript**. The javascript objects are string, array, date etc.

#### Note: if html document contains frame or iframe, browser creates additional window objects for each frame.

## Methods of window object

The important methods of window object are as follows:

|  |  |
| --- | --- |
| **Method** | **Description** |
| alert() | displays the alert box containing message with ok button. |
| confirm() | displays the confirm dialog box containing message with ok and cancel button. |
| prompt() | displays a dialog box to get input from the user. |
| open() | opens the new window. |
| close() | closes the current window. |
| setTimeout() | performs action after specified time like calling function, evaluating expressions etc. |

#### Example of alert() in javascript

It displays alert dialog box. It has message and ok button.

1. <script type="text/javascript">
2. function msg(){
3. alert("Hello Alert Box");
4. }
5. </script>
6. <input type="button" value="click" onclick="msg()"/>

**Example of confirm() in javascript**

It displays the confirm dialog box. It has message with ok and cancel buttons.

1. <script type="text/javascript">
2. function msg(){
3. var v= confirm("Are u sure?");
4. if(v==true){
5. alert("ok");
6. }
7. else{
8. alert("cancel");
9. }
11. }
12. </script>
14. <input type="button" value="delete record" onclick="msg()"/>

**Example of prompt() in javascript**

It displays prompt dialog box for input. It has message and textfield.

1. <script type="text/javascript">
2. function msg(){
3. var v= prompt("Who are you?");
4. alert("I am "+v);
6. }
7. </script>
9. <input type="button" value="click" onclick="msg()"/>

**Example of open() in javascript**

It displays the content in a new window.

1. <script type="text/javascript">
2. function msg(){
3. open("http://www.javatpoint.com");
4. }
5. </script>
6. <input type="button" value="javatpoint" onclick="msg()"/>

**Example of setTimeout() in javascript**

It performs its task after the given milliseconds.

1. <script type="text/javascript">
2. function msg(){
3. setTimeout(
4. function(){
5. alert("Welcome to Javatpoint after 2 seconds")
6. },2000);
8. }
9. </script>
11. <input type="button" value="click" onclick="msg()"/>

# Document Object Model

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

When html document is loaded in the browser, it becomes a document object. It is the **root element** that represents the html document.

As mentioned earlier, it is the object of window. So

window.document

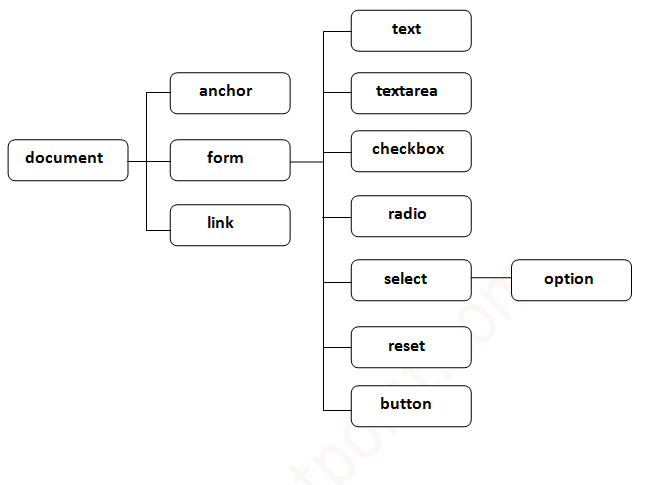
Is same as

document

According to W3C - *"The W3C Document Object Model (DOM) is a platform and language-neutral interface that allows* [*programs[http://cdncache1-a.akamaihd.net/items/it/img/arrow-10x10.png](http://www.javatpoint.com/document-object-model)*](http://www.javatpoint.com/document-object-model) *and scripts to dynamically access and update the content, structure, and style of a document."*

## Properties of document object

Let's see the properties of document object that can be accessed and modified by the document object.



## Methods of document object

We can access and change the contents of document by its methods.

The important methods of document object are as follows:

|  |  |
| --- | --- |
| **Method** | **Description** |
| write("string") | writes the given string on the doucment. |
| writeln("string") | writes the given string on the doucment with newline character at the end. |
| getElementById() | returns the element having the given id value. |
| getElementsByName() | returns all the elements having the given name value. |
| getElementsByTagName() | returns all the elements having the given tag name. |
| getElementsByClassName() | returns all the elements having the given class name. |

### Accessing the field value by document object

In this example, we are going to get the value of input text by user. Here, we are using **document.form1.name.value** to get the value of name field.

Here, **document** is the root element that represents the html document.

**form1** is the name of the form.

**name** is the attribute name of the input text.

**value** is the property, that returns the value of the input text.

Let's see the simple example of document object that prints name with welcome message.

1. <script type="text/javascript">
2. function printvalue(){
3. var name=document.form1.name.value;
4. alert("Welcome: "+name);
5. }
6. </script>
8. <form name="form1">
9. Enter Name:<input type="text" name="name"/>
10. <input type="button" onclick="printvalue()" value="print name"/>
11. </form>

# Javascript - document.getElementById() method

The **document.getElementById()** [method[http://cdncache1-a.akamaihd.net/items/it/img/arrow-10x10.png](http://www.javatpoint.com/document-getElementById()-method)](http://www.javatpoint.com/document-getElementById%28%29-method) returns the element of specified id.

In the previous page, we have used **document.form1.name.value** to get the value of the input value. Instead of this, we can use document.getElementById() method to get value of the input text. But we need to define id for the input field.

Let's see the simple example of document.getElementById() method that prints cube of the given number.

1. <script type="text/javascript">
2. function getcube(){
3. var number=document.getElementById("number").value;
4. alert(number\*number\*number);
5. }
6. </script>
7. <form>
8. Enter No:<input type="text" id="number" name="number"/><br/>
9. <input type="button" value="cube" onclick="getcube()"/>
10. </form>

# Javascript - document.getElementsByName() method

The **document.getElementsByName()** method returns all the element of specified name.

The syntax of the getElementsByName() method is given below:

1. document.getElementsByName("name")

Here, name is required.

### Example of document.getElementsByName() method

In this example, we going to count total number of genders. Here, we are using getElementsByName() method to get all the genders.

1. <script type="text/javascript">
2. function totalelements()
3. {
4. var allgenders=document.getElementsByName("gender");
5. alert("Total Genders:"+allgenders.length);
6. }
7. </script>
8. <form>
9. Male:<input type="radio" name="gender" value="male">
10. Female:<input type="radio" name="gender" value="female">
12. <input type="button" onclick="totalelements()" value="Total Genders">
13. </form>

# Javascript - document.getElementsByTagName() method

The **document.getElementsByTagName()** [method[http://cdncache1-a.akamaihd.net/items/it/img/arrow-10x10.png](http://www.javatpoint.com/document-getElementsByTagName()-method)](http://www.javatpoint.com/document-getElementsByTagName%28%29-method) returns all the element of specified tag name.

The syntax of the getElementsByTagName() method is given below:

1. document.getElementsByTagName("name")

Here, name is required.

### Example of document.getElementsByTagName() method

In this example, we going to count total number of paragraphs used in the document. To do this, we have called the document.getElementsByTagName("p") method that returns the total paragraphs.

1. <script type="text/javascript">
2. function countpara(){
3. var totalpara=document.getElementsByTagName("p");
4. alert("total p tags are: "+totalpara.length);
6. }
7. </script>
8. <p>This is a pragraph</p>
9. <p>Here we are going to count total number of paragraphs by getElementByTagName() [method[http://cdncache1-a.akamaihd.net/items/it/img/arrow-10x10.png](http://www.javatpoint.com/document-getElementsByTagName()-method)](http://www.javatpoint.com/document-getElementsByTagName%28%29-method).</p>
10. <p>Let's see the simple example</p>
11. <button onclick="countpara()">count paragraph</button>

### Another example of document.getElementsByTagName() method

In this example, we going to count total number of h2 and h3 tags used in the document.

1. <script type="text/javascript">
2. function counth2(){
3. var totalh2=document.getElementsByTagName("h2");
4. alert("total h2 tags are: "+totalh2.length);
5. }
6. function counth3(){
7. var totalh3=document.getElementsByTagName("h3");
8. alert("total h3 tags are: "+totalh3.length);
9. }
10. </script>
11. <h2>This is h2 tag</h2>
12. <h2>This is h2 tag</h2>
13. <h3>This is h3 tag</h3>
14. <h3>This is h3 tag</h3>
15. <h3>This is h3 tag</h3>
16. <button onclick="counth2()">count h2</button>
17. <button onclick="counth3()">count h3</button>

#### Note: Output of the given examples may differ on this page because it will count the total number of para , total number of h2 and total number of h3 tags used in this document.

# Javascript - innerHTML

The **innerHTML** property can be used to write the dynamic html on the html document.

It is used mostly in the web pages to generate the dynamic html such as [registration form[http://cdncache1-a.akamaihd.net/items/it/img/arrow-10x10.png](http://www.javatpoint.com/javascript-innerHTML)](http://www.javatpoint.com/javascript-innerHTML), comment form, links etc.

### Example of innerHTML property

In this example, we are going to create the html form when user clicks on the button.

In this example, we are dynamically [writing[http://cdncache1-a.akamaihd.net/items/it/img/arrow-10x10.png](http://www.javatpoint.com/javascript-innerHTML)](http://www.javatpoint.com/javascript-innerHTML) the html form inside the div name having the id mylocation. We are identifing this position by calling the document.getElementById() [method[http://cdncache1-a.akamaihd.net/items/it/img/arrow-10x10.png](http://www.javatpoint.com/javascript-innerHTML)](http://www.javatpoint.com/javascript-innerHTML).

1. <script type="text/javascript" >
2. function showcommentform() {
3. var data="Name:<input type='text' name='name'><br>Comment:<textarea rows='5' cols='80'></textarea><br><input type='submit' value='comment'>";
5. document.getElementById('mylocation').innerHTML=data;
6. }
8. </script>
9. <form name="myForm">
10. <input type="button" value="comment" onclick="showcommentform()">
11. <div id="mylocation"></div>
12. </form>

# Javascript - innerText

The **innerText** property can be used to write the dynamic text on the html document. Here, text will not be interpreted as html text but a normal text.

It is used mostly in the web pages to generate the dynamic content such as [writing[http://cdncache1-a.akamaihd.net/items/it/img/arrow-10x10.png](http://www.javatpoint.com/javascript-innerText)](http://www.javatpoint.com/javascript-innerText) the validation message, password strength etc.

### Example of innerText property

In this example, we are going to display the password strength when releases the key after press.

1. <script type="text/javascript" >
2. function validate() {
3. var msg;
4. if(document.myForm.userPass.value.length>5){
5. msg="good";
6. }
7. else{
8. msg="poor";
9. }
10. document.getElementById('mylocation').innerText=msg;
11. }
13. </script>
14. <form name="myForm">
16. <input type="password" value="" name="userPass" onkeyup="validate()">
17. Strength:<span id="mylocation">no strength</span>
19. </form>

# Javascript Form Validation

It is important to validate the form submitted by the user because it can have inappropriate values. So validation is must.

The javascript provides you the facility the validate the form on the client side so processing will be fast than server-side validation. So, most of the web developers prefer javascript validation.

Through javascript, we can validate name, password, email, date, mobile number etc fields.

### Example of javascript form validation

In this example, we are going to validate the name and password. The name can’t be empty and password can’t be less than 6 characters long.

Here, we are validating the form on form submit. The user will not be forwarded to the next page until given values are correct.

1. <script>
2. function validateform(){
3. var name=document.myform.name.value;
4. var password=document.myform.password.value;
6. if (name==null || name==""){
7. alert("Name can't be blank");
8. return false;
9. }else if(password.length<6){
10. alert("Password must be at least 6 characters long.");
11. return false;
12. }
13. }
14. </script>
15. <body>
16. <form name="myform" method="post" action="abc.jsp" onsubmit="return validateform()" >
17. Name: <input type="text" name="name"><br/>
18. Password: <input type="password" name="password"><br/>
19. <input type="submit" value="[register[http://cdncache1-a.akamaihd.net/items/it/img/arrow-10x10.png](http://www.javatpoint.com/javascript-form-validation)](http://www.javatpoint.com/javascript-form-validation)">
20. </form>

### Javascript email validation

We can validate the email by the help of javascript.

There are many criteria that need to be follow to validate the email id such as:

* email id must contain the @ and . character
* There must be at least one character before and after the @.
* There must be at least two characters after . (dot).

Let's see the simple example to validate the email field.

1. <script>
2. function validateemail()
3. {
4. var x=document.myform.email.value;
5. var atposition=x.indexOf("@");
6. var dotposition=x.lastIndexOf(".");
7. if (atposition<1 || dotposition<atposition+2 || dotposition+2>=x.length){
8. alert("Please enter a valid e-mail address \n atpostion:"+atposition+"\n dotposition:"+dotposition);
9. return false;
10. }
11. }
12. </script>
13. <body>
14. <form name="myform"  method="post" action="abc.jsp" onsubmit="return validateemail();">
15. Email: <input type="text" name="email"><br/>
17. <input type="submit" value="[register[http://cdncache1-a.akamaihd.net/items/it/img/arrow-10x10.png](http://www.javatpoint.com/javascript-form-validation)](http://www.javatpoint.com/javascript-form-validation)">
18. </form>

<http://www.javatpoint.com/javascript-tutorial>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_