JavaScript Example

1. [JavaScript Example](https://www.javatpoint.com/javascript-example)
2. [Within body tag](https://www.javatpoint.com/javascript-example)
3. [Within head tag](https://www.javatpoint.com/javascript-example)

Javascript example is easy to code. JavaScript provides 3 places to put the JavaScript code: within body tag, within head tag and external JavaScript file.

Let’s create the first JavaScript example.

1. **<script** type="text/javascript"**>**
2. document.write("JavaScript is a simple language for javatpoint learners");
3. **</script>**

[**Test it Now**](https://www.javatpoint.com/oprweb/test.jsp?filename=example1js)

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.

Play Videox[](https://campaign.adpushup.com/get-started/?utm_source=banner&utm_campaign=growth_hack)

The **document.write()** function is used to display dynamic content through JavaScript. We will learn about document object in detail later.

3 Places to put JavaScript code

1. Between the body tag of html
2. Between the head tag of html
3. In .js file (external javaScript)

1) JavaScript Example : code between the body tag

In the above example, we have displayed the dynamic content using JavaScript. Let’s see the simple example of JavaScript that displays alert dialog box.

1. **<script** type="text/javascript"**>**
2. alert("Hello Javatpoint");
3. **</script>**

[**Test it Now**](https://www.javatpoint.com/oprweb/test.jsp?filename=example2js)

2) JavaScript Example : code between the head tag

Let’s see the same example of displaying alert dialog box of JavaScript that is contained inside the head tag.

In this example, we are creating a function msg(). To create function in JavaScript, you need to write function with function\_name 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 external JavaScript file and embed it in many html page.

It provides **code re usability** because single JavaScript file can be used in several html pages.

An external JavaScript file must be saved by .js extension. It is recommended to embed all JavaScript files into a single file. It increases the speed of the webpage.

Let's create an external [JavaScript](https://www.javatpoint.com/javascript-tutorial) file that prints Hello Javatpoint in a alert dialog box.

Play Videox[](https://campaign.adpushup.com/get-started/?utm_source=banner&utm_campaign=growth_hack)

**message.js**

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

Let's include the JavaScript file into [html](https://www.javatpoint.com/html-tutorial) page. It calls the [JavaScript function](https://www.javatpoint.com/javascript-function) on button click.

**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>**

Advantages of External JavaScript

There will be following benefits if a user creates an external javascript:

1. It helps in the reusability of code in more than one HTML file.
2. It allows easy code readability.
3. It is time-efficient as web browsers cache the external js files, which further reduces the page loading time.
4. It enables both web designers and coders to work with html and js files parallelly and separately, i.e., without facing any code conflictions.
5. The length of the code reduces as only we need to specify the location of the js file.

# JavaScript Comment

1. [JavaScript comments](https://www.javatpoint.com/javascript-comment)
2. [Advantage of javaScript comments](https://www.javatpoint.com/javascript-comment)
3. [Single-line and Multi-line comments](https://www.javatpoint.com/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 end user can easily interpret the code.

The JavaScript comment is ignored by the JavaScript engine i.e. embedded in the browser.

#### Advantages of JavaScript comments

There are mainly two advantages of JavaScript comments.

Play Videox[](https://campaign.adpushup.com/get-started/?utm_source=banner&utm_campaign=growth_hack)

1. **To make code easy to understand** It can be used to elaborate the code so that end user can easily understand the code.
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 JavaScript.

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

## JavaScript Single line Comment

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

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

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

[**Test it Now**](https://www.javatpoint.com/oprweb/test.jsp?filename=comment1js)

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>**

[**Test it Now**](https://www.javatpoint.com/oprweb/test.jsp?filename=comment2js)

## JavaScript 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

1. [JavaScript variable](https://www.javatpoint.com/javascript-variable)
2. [JavaScript Local variable](https://www.javatpoint.com/javascript-variable#local)
3. [JavaScript Global variable](https://www.javatpoint.com/javascript-variable#gloabl)

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

There are some rules while declaring a JavaScript 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.

## Correct JavaScript variables

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

## Incorrect JavaScript variables

1. var  123=30;
2. var \*aa=320;

## Example of JavaScript variable

Let’s see a simple example of JavaScript variable.

Play Videox[](https://campaign.adpushup.com/get-started/?utm_source=banner&utm_campaign=growth_hack)

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

[**Test it Now**](https://www.javatpoint.com/oprweb/test.jsp?filename=jsvariable1)

#### Output of the above example

30

## JavaScript local variable

A JavaScript local variable is declared inside block or function. 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 **JavaScript 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:

1. **<script>**
2. var data=200;//gloabal variable
3. function a(){
4. document.writeln(data);
5. }
6. function b(){
7. document.writeln(data);
8. }
9. a();//calling JavaScript function
10. b();
11. **</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

JavaScript primitive data types

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

|  |  |
| --- | --- |
| **Data Type** | **Description** |
| String | represents sequence of characters e.g. "hello" |
| Number | represents numeric values e.g. 100 |
| Boolean | represents boolean value either false or true |
| Undefined | represents undefined value |
| Null | represents null i.e. no value at all |

JavaScript non-primitive data types

The non-primitive data types are as follows:

Play Video

|  |  |
| --- | --- |
| **Data Type** | **Description** |
| Object | represents instance through which we can access members |
| Array | represents group of similar values |
| RegExp | represents regular expression |

JavaScript Operators

JavaScript operators are symbols that 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

JavaScript Arithmetic Operators

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

Play Video[](https://campaign.adpushup.com/get-started/?utm_source=banner&utm_campaign=growth_hack)

|  |  |  |
| --- | --- | --- |
| **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 |

# JavaScript Functions

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

#### Advantage of JavaScript function

There are mainly two advantages of JavaScript functions.

1. **Code reusability**: We can call a function several times so it save coding.
2. **Less coding**: It makes our program compact. We don’t need to write many lines of code each time to perform a common task.

## JavaScript Function Syntax

The syntax of declaring function is given below.

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

JavaScript Functions can have 0 or more arguments.

Play Videox

## JavaScript Function Example

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"**/>**

[**Test it Now**](https://www.javatpoint.com/oprweb/test.jsp?filename=function1js)

#### Output of the above example

## JavaScript 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>**

[**Test it Now**](https://www.javatpoint.com/oprweb/test.jsp?filename=function2js)

#### Output of the above example

Top of Form

Bottom of Form

## Function with Return Value

We can call function that returns a value and use it in our program. Let’s see the example of function that returns value.

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

[**Test it Now**](https://www.javatpoint.com/oprweb/test.jsp?filename=function3js)

#### Output of the above example

hello javatpoint! How r u?

## JavaScript Function Object

In JavaScript, the purpose of **Function constructor** is to create a new Function object. It executes the code globally. However, if we call the constructor directly, a function is created dynamically but in an unsecured way.

## Syntax

1. new Function ([arg1[, arg2[, ....argn]],] functionBody)

## Parameter

**arg1, arg2, .... , argn** - It represents the argument used by function.

**functionBody** - It represents the function definition.

## JavaScript Function Methods

Let's see function methods with description.

|  |  |
| --- | --- |
| **Method** | **Description** |
| [apply()](https://www.javatpoint.com/javascript-function-apply-method) | It is used to call a function contains this value and a single array of arguments. |
| [bind()](https://www.javatpoint.com/javascript-function-bind-method) | It is used to create a new function. |
| [call()](https://www.javatpoint.com/javascript-function-call-method) | It is used to call a function contains this value and an argument list. |
| [toString()](https://www.javatpoint.com/javascript-function-tostring-method) | It returns the result in a form of a string. |

## JavaScript Function Object Examples

### Example 1

Let's see an example to display the sum of given numbers.

1. **<script>**
2. var add=new Function("num1","num2","return num1+num2");
3. document.writeln(add(2,5));
4. **</script>**

[**Test it Now**](https://www.javatpoint.com/oprweb/test.jsp?filename=JavaScriptFunctionObjectExample1)

**Output:**

7

### Example 2

Let's see an example to display the power of provided value.

1. **<script>**
2. var pow=new Function("num1","num2","return Math.pow(num1,num2)");
3. document.writeln(pow(2,3));
4. **</script>**

# JavaScript Objects

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 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.

Play Videox

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

## 1) JavaScript 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).

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>**

[**Test it Now**](https://www.javatpoint.com/oprweb/test.jsp?filename=object1js)

#### Output of the above example

102 Shyam Kumar 40000

## 2) By creating instance of Object

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>**

[**Test it Now**](https://www.javatpoint.com/oprweb/test.jsp?filename=object2js)

#### Output of the above example

101 Ravi 50000

## 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>**

[**Test it Now**](https://www.javatpoint.com/oprweb/test.jsp?filename=object3js)

#### Output of the above example

103 Vimal Jaiswal 30000

JavaScript String

The **JavaScript string** is an object that represents a sequence of characters.

There are 2 ways to create string in JavaScript

1. By string literal
2. By string object (using new keyword)

1) By string literal

The string literal is created using double quotes. The syntax of creating string using string literal is given below:

1. var stringname="string value";

Let's see the simple example of creating string literal.

Play Videox[](https://campaign.adpushup.com/get-started/?utm_source=banner&utm_campaign=growth_hack)

1. **<script>**
2. var str="This is string literal";
3. document.write(str);
4. **</script>**

[**Test it Now**](https://www.javatpoint.com/oprweb/test.jsp?filename=jsstring1)

**Output:**

This is string literal

2) By string object (using new keyword)

The syntax of creating string object using new keyword is given below:

1. var stringname=new String("string literal");

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

Let's see the example of creating string in JavaScript by new keyword.

1. **<script>**
2. var stringname=new String("hello javascript string");
3. document.write(stringname);
4. **</script>**

[**Test it Now**](https://www.javatpoint.com/oprweb/test.jsp?filename=jsstring2)

**Output:**

hello javascript string

JavaScript String Methods

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

|  |  |
| --- | --- |
| **Methods** | **Description** |
| [charAt()](https://www.javatpoint.com/javascript-string-charat-method) | It provides the char value present at the specified index. |
| [charCodeAt()](https://www.javatpoint.com/javascript-string-charcodeat-method) | It provides the Unicode value of a character present at the specified index. |
| [concat()](https://www.javatpoint.com/javascript-string-concat-method) | It provides a combination of two or more strings. |
| [indexOf()](https://www.javatpoint.com/javascript-string-indexof-method) | It provides the position of a char value present in the given string. |
| [lastIndexOf()](https://www.javatpoint.com/javascript-string-lastindexof-method) | It provides the position of a char value present in the given string by searching a character from the last position. |
| [search()](https://www.javatpoint.com/javascript-string-search-method) | It searches a specified regular expression in a given string and returns its position if a match occurs. |
| [match()](https://www.javatpoint.com/javascript-string-match-method) | It searches a specified regular expression in a given string and returns that regular expression if a match occurs. |
| [replace()](https://www.javatpoint.com/javascript-string-replace-method) | It replaces a given string with the specified replacement. |
| [substr()](https://www.javatpoint.com/javascript-string-substr-method) | It is used to fetch the part of the given string on the basis of the specified starting position and length. |
| [substring()](https://www.javatpoint.com/javascript-string-substring-method) | It is used to fetch the part of the given string on the basis of the specified index. |
| [slice()](https://www.javatpoint.com/javascript-string-slice-method) | It is used to fetch the part of the given string. It allows us to assign positive as well negative index. |
| [toLowerCase()](https://www.javatpoint.com/javascript-string-tolowercase-method) | It converts the given string into lowercase letter. |
| [toLocaleLowerCase()](https://www.javatpoint.com/javascript-string-tolocalelowercase-method) | It converts the given string into lowercase letter on the basis of host?s current locale. |
| [toUpperCase()](https://www.javatpoint.com/javascript-string-touppercase-method) | It converts the given string into uppercase letter. |
| [toLocaleUpperCase()](https://www.javatpoint.com/javascript-string-tolocaleuppercase-method) | It converts the given string into uppercase letter on the basis of host?s current locale. |
| [toString()](https://www.javatpoint.com/javascript-string-tostring-method) | It provides a string representing the particular object. |
| [valueOf()](https://www.javatpoint.com/javascript-string-valueof-method) | It provides the primitive value of string object. |
| split() | It splits a string into substring array, then returns that newly created array. |
| trim() | It trims the white space from the left and right side of the string. |

1) JavaScript String charAt(index) Method

The JavaScript String charAt() method returns the character at the given index.

1. **<script>**
2. var str="javascript";
3. document.write(str.charAt(2));
4. **</script>**

# Browser Object Model

1. [Browser Object Model (BOM)](https://www.javatpoint.com/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");

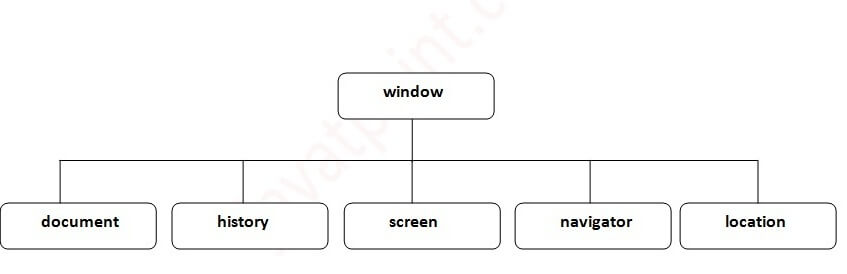
is same as:

1. alert("hello javatpoint");

You can use a lot of properties (other objects) defined underneath the window object like document, history, screen, navigator, location, innerHeight, innerWidth,

Play Videox[](https://campaign.adpushup.com/get-started/?utm_source=banner&utm_campaign=growth_hack)

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



Visit the next page to learn about window object fully with example.

# Window Object

1. [Window Object](https://www.javatpoint.com/window-object)
2. [Properties of Window Object](https://www.javatpoint.com/window-object)
3. [Methods of Window Object](https://www.javatpoint.com/window-object)
4. [Example of Window Object](https://www.javatpoint.com/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.

Play Videox[](https://campaign.adpushup.com/get-started/?utm_source=banner&utm_campaign=growth_hack)

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

#### Output of the above example

#### 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()"**/>**

#### Output of the above example

#### 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()"**/>**

#### Output of the above example

#### 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()"**/>**

#### Output of the above example

#### 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()"**/>**

#### Output of the above example

# Window Object

1. [Window Object](https://www.javatpoint.com/window-object)
2. [Properties of Window Object](https://www.javatpoint.com/window-object)
3. [Methods of Window Object](https://www.javatpoint.com/window-object)
4. [Example of Window Object](https://www.javatpoint.com/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.

Play Videox[](https://campaign.adpushup.com/get-started/?utm_source=banner&utm_campaign=growth_hack)

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

#### Output of the above example

#### 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()"**/>**

#### Output of the above example

#### 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()"**/>**

#### Output of the above example

#### 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()"**/>**

#### Output of the above example

#### 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()"**/>**

#### Output of the above example

# JavaScript Screen Object

1. [Screen Object](https://www.javatpoint.com/javascript-screen)
2. [Properties of Screen Object](https://www.javatpoint.com/javascript-screen)
3. [Methods of Screen Object](https://www.javatpoint.com/javascript-screen)
4. [Example of Screen Object](https://www.javatpoint.com/javascript-screen)

The **JavaScript screen object** holds information of browser screen. It can be used to display screen width, height, colorDepth, pixelDepth etc.

The navigator object is the window property, so it can be accessed by:

1. window.screen

Or,

1. screen

## Property of JavaScript Screen Object

There are many properties of screen object that returns information of the browser.

9.7M

92

Competitive questions on Structures

|  |  |  |
| --- | --- | --- |
| **No.** | **Property** | **Description** |
| 1 | width | returns the width of the screen |
| 2 | height | returns the height of the screen |
| 3 | availWidth | returns the available width |
| 4 | availHeight | returns the available height |
| 5 | colorDepth | returns the color depth |
| 6 | pixelDepth | returns the pixel depth. |

#### Example of JavaScript Screen Object

Let’s see the different usage of screen object.

1. **<script>**
2. document.writeln("**<br/>**screen.width: "+screen.width);
3. document.writeln("**<br/>**screen.height: "+screen.height);
4. document.writeln("**<br/>**screen.availWidth: "+screen.availWidth);
5. document.writeln("**<br/>**screen.availHeight: "+screen.availHeight);
6. document.writeln("**<br/>**screen.colorDepth: "+screen.colorDepth);
7. document.writeln("**<br/>**screen.pixelDepth: "+screen.pixelDepth);
8. **</script>**

[**Test it Now**](https://www.javatpoint.com/oprweb/test.jsp?filename=jsscreen)

screen.width: 1366

screen.height: 768

screen.availWidth: 1366

screen.availHeight: 728

screen.colorDepth: 24

screen.pixelDepth: 24

# Document Object Model

1. [Document Object](https://www.javatpoint.com/document-object-model)
2. [Properties of document object](https://www.javatpoint.com/document-object-model)
3. [Methods of document object](https://www.javatpoint.com/document-object-model)
4. [Example of document object](https://www.javatpoint.com/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. It has properties and methods. By the help of document object, we can add dynamic content to our web page.

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

1. window.document

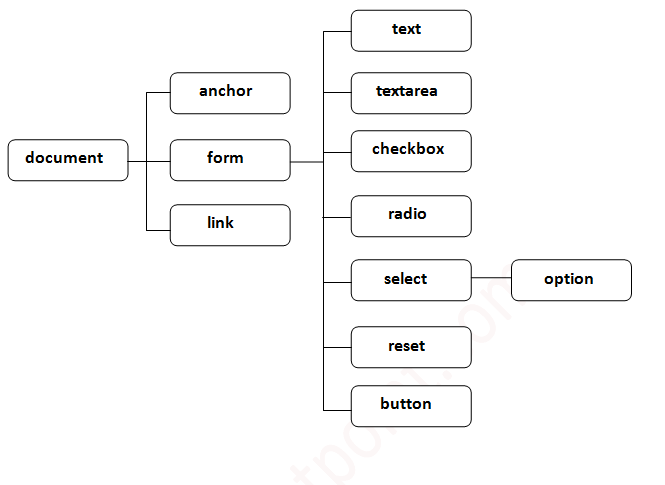
Is same as

Play Videox[](https://campaign.adpushup.com/get-started/?utm_source=banner&utm_campaign=growth_hack)

1. document

According to W3C - *"The W3C Document Object Model (DOM) is a platform and language-neutral interface that allows programs 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 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>**

#### Output of the above example

# Javascript - document.getElementById() method

1. [getElementById() method](https://www.javatpoint.com/document-getElementById()-method)
2. [Example of getElementById()](https://www.javatpoint.com/document-getElementById()-method)

The **document.getElementById()** 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.

Play Videox[](https://campaign.adpushup.com/get-started/?utm_source=banner&utm_campaign=growth_hack)

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>**

#### Output of the above example

Top of Form

Enter No:

Bottom of Form

# Javascript - document.getElementsByName() method

1. [getElementsByName() method](https://www.javatpoint.com/document-getElementsByName()-method)
2. [Example of getElementsByName()](https://www.javatpoint.com/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.

Play Videox[](https://campaign.adpushup.com/get-started/?utm_source=banner&utm_campaign=growth_hack)

### 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>**

#### Output of the above example

# Javascript - document.getElementsByTagName() method

1. [getElementsByTagName() method](https://www.javatpoint.com/document-getElementsByTagName()-method)
2. [Example of getElementsByTagName()](https://www.javatpoint.com/document-getElementsByTagName()-method)

The **document.getElementsByTagName()** 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.

Play Videox[](https://campaign.adpushup.com/get-started/?utm_source=banner&utm_campaign=growth_hack)

### 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.**</p>**
10. **<p>**Let's see the simple example**</p>**
11. **<button** onclick="countpara()"**>**count paragraph**</button>**

#### Output of the above example

This is a pragraph

Here we are going to count total number of paragraphs by getElementByTagName() method.

Let's see the simple example

count paragraph

### 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>**

# Javascript - innerHTML

1. [javascript innerHTML](https://www.javatpoint.com/javascript-innerHTML)
2. [Example of innerHTML property](https://www.javatpoint.com/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, 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.

Play Videox[](https://campaign.adpushup.com/get-started/?utm_source=banner&utm_campaign=growth_hack)

In this example, we are dynamically writing the html form inside the div name having the id mylocation. We are identifing this position by calling the document.getElementById() method.

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

[**Test it Now**](https://www.javatpoint.com/oprweb/test.jsp?filename=jsinnerhtml)

#### Output of the above example

Top of Form

Bottom of Form

### Show/Hide Comment Form Example using innerHTML

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**First JS**</title>**
5. **<script>**
6. var flag=true;
7. function commentform(){
8. var cform="**<form** action='Comment'**>**Enter Name:**<br><input** type='text' name='name'**/><br/>**
9. Enter Email:**<br><input** type='email' name='email'**/><br>**Enter Comment:**<br/>**
10. **<textarea** rows='5' cols='70'**></textarea><br><input** type='submit' value='Post Comment'**/></form>**";
11. if(flag){
12. document.getElementById("mylocation").innerHTML=cform;
13. flag=false;
14. }else{
15. document.getElementById("mylocation").innerHTML="";
16. flag=true;
17. }
18. }
19. **</script>**
20. **</head>**
21. **<body>**
22. **<button** onclick="commentform()"**>**Comment**</button>**
23. **<div** id="mylocation"**></div>**
24. **</body>**
25. **</html>**

JavaScript Form Validation

1. [JavaScript form validation](https://www.javatpoint.com/javascript-form-validation)
2. [Example of JavaScript validation](https://www.javatpoint.com/javascript-form-validation)
3. [JavaScript email validation](https://www.javatpoint.com/javascript-form-validation#email)

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

JavaScript provides facility to validate the form on the client-side so data processing will be faster than server-side validation. Most of the web developers prefer JavaScript form validation.

Through JavaScript, we can validate name, password, email, date, mobile numbers and more fields.

JavaScript Form Validation Example

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.

Play Videox[](https://campaign.adpushup.com/get-started/?utm_source=banner&utm_campaign=growth_hack)

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"**>**
20. **</form>**

[**Test it Now**](https://www.javatpoint.com/oprweb/test.jsp?filename=jsvalidation1)

JavaScript Retype Password Validation

1. **<script** type="text/javascript"**>**
2. function matchpass(){
3. var firstpassword=document.f1.password.value;
4. var secondpassword=document.f1.password2.value;
6. if(firstpassword==secondpassword){
7. return true;
8. }
9. else{
10. alert("password must be same!");
11. return false;
12. }
13. }
14. **</script>**
16. **<form** name="f1" action="register.jsp" onsubmit="return matchpass()"**>**
17. Password:**<input** type="password" name="password" **/><br/>**
18. Re-enter Password:**<input** type="password" name="password2"**/><br/>**
19. **<input** type="submit"**>**
20. **</form>**

[**Test it Now**](https://www.javatpoint.com/oprweb/test.jsp?filename=jsvalidation4)

JavaScript Number Validation

Let's validate the textfield for numeric value only. Here, we are using isNaN() function.

1. **<script>**
2. function validate(){
3. var num=document.myform.num.value;
4. if (isNaN(num)){
5. document.getElementById("numloc").innerHTML="Enter Numeric value only";
6. return false;
7. }else{
8. return true;
9. }
10. }
11. **</script>**
12. **<form** name="myform" onsubmit="return validate()" **>**
13. Number: **<input** type="text" name="num"**><span** id="numloc"**></span><br/>**
14. **<input** type="submit" value="submit"**>**
15. **</form>**

[**Test it Now**](https://www.javatpoint.com/oprweb/test.jsp?filename=jsvalidation5)

JavaScript validation with image

Let’s see an interactive JavaScript form validation example that displays correct and incorrect image if input is correct or incorrect.

1. **<script>**
2. function validate(){
3. var name=document.f1.name.value;
4. var password=document.f1.password.value;
5. var status=false;
7. if(name.length**<1**){
8. document.getElementById("nameloc").innerHTML=
9. " <img src='unchecked.gif'/> Please enter your name";
10. status=false;
11. }else{
12. document.getElementById("nameloc").innerHTML=" <img src='checked.gif'/>";
13. status=true;
14. }
15. if(password.length**<6**){
16. document.getElementById("passwordloc").innerHTML=
17. " <img src='unchecked.gif'/> Password must be at least 6 char long";
18. status=false;
19. }else{
20. document.getElementById("passwordloc").innerHTML=" <img src='checked.gif'/>";
21. }
22. return status;
23. }
24. **</script>**
26. **<form** name="f1" action="#" onsubmit="return validate()"**>**
27. **<table>**
28. **<tr><td>**Enter Name:**</td><td><input** type="text" name="name"**/>**
29. **<span** id="nameloc"**></span></td></tr>**
30. **<tr><td>**Enter Password:**</td><td><input** type="password" name="password"**/>**
31. **<span** id="passwordloc"**></span></td></tr>**
32. **<tr><td** colspan="2"**><input** type="submit" value="register"**/></td></tr>**
33. **</table>**
34. **</form>**

[**Test it Now**](https://www.javatpoint.com/oprweb/test.jsp?filename=jsvalidation2)

Output:

Top of Form

|  |  |
| --- | --- |
| Enter Name: |  |
| Enter Password: |  |
|  | |

Bottom of 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="#" onsubmit="return validateemail();"**>**
15. Email: **<input** type="text" name="email"**><br/>**
17. **<input** type="submit" value="register"**>**
18. **</form>**