**JAVA SCRIPT:**

## typeof Operator

The typeof operator is a unary operator that is placed before its single operand, which can be of any type. Its value is a string indicating the data type of the operand.

The typeof operator evaluates to "number", "string", or "boolean" if its operand is a number, string, or boolean value and returns true or false based on the evaluation.

**result = (typeof b == "string" ? "B is String" : "B is Numeric");**

# preventDefault() Event Method

## Definition and Usage

The preventDefault() method cancels the event if it is cancelable, meaning that the default action that belongs to the event will not occur.

For example, this can be useful when:

Clicking on a "Submit" button, prevent it from submitting a form

Clicking on a link, prevent the link from following the URL

**Note:** Not all events are cancelable. Use the [cancelable](http://www.w3schools.com/jsref/event_cancelable.asp) property to find out if an event is cancelable.

**Note:** The preventDefault() method does not prevent further propagation of an event through the DOM. Use the stopPropagation() method to handle this.

Prevent the default action of a checkbox:

document.getElementById("myCheckbox").addEventListener("click",

function(event){  
    event.preventDefault()  
});

1. The ‘**with’** keyword is used as a kind of shorthand for referencing an object's properties or methods.

The object specified as an argument to with becomes the default object for the duration of the block that follows. The properties and methods for the object can be used without naming the object.

1. getElementById( id)

Returns the Element of this document that has the specified value for its id attribute, or null if no such Element exists in the document.

**Ex − document.getElementById( id)**

1. getElementsByName( name)

Returns an array of nodes of all elements in the document that have a specified value for their name attribute. If no such elements are found, returns a zero-length array.

**Ex − document.getElementsByName( name)**

1. getElementsByTagName( tagname)

Returns an array of all Element nodes in this document that have the specified tag name. The Element nodes appear in the returned array in the same order they appear in the document source.

Ex **− document.getElementsByTagName( tagname)**

Every web page resides inside a browser window which can be considered as an object.

A Document object represents the HTML document that is displayed in that window. The Document object has various properties that refer to other objects which allow access to and modification of document content.

The way a document content is accessed and modified is called the **Document Object Model**, or **DOM**. The Objects are organized in a hierarchy. This hierarchical structure applies to the organization of objects in a Web document.

* **Window object** − Top of the hierarchy. It is the outmost element of the object hierarchy.
* **Document object** − Each HTML document that gets loaded into a window becomes a document object. The document contains the contents of the page.
* **Form object** − Everything enclosed in the <form>...</form> tags sets the form object.
* **Form control elements** − The form object contains all the elements defined for that object such as text fields, buttons, radio buttons, and checkboxes.

Here is a simple hierarchy of a few important objects −



There are several DOMs in existence. The following sections explain each of these DOMs in detail and describe how you can use them to access and modify document content.

* [The Legacy DOM](https://www.tutorialspoint.com/javascript/javascript_legacy_dom.htm) − This is the model which was introduced in early versions of JavaScript language. It is well supported by all browsers, but allows access only to certain key portions of documents, such as forms, form elements, and images.
* [The W3C DOM](https://www.tutorialspoint.com/javascript/javascript_w3c_dom.htm) − This document object model allows access and modification of all document content and is standardized by the World Wide Web Consortium (W3C). This model is supported by almost all the modern browsers.
* [The IE4 DOM](https://www.tutorialspoint.com/javascript/javascript_ie4_dom.htm) − This document object model was introduced in Version 4 of Microsoft's Internet Explorer browser. IE 5 and later versions include support for most basic W3C DOM features.

## DOM compatibility

If you want to write a script with the flexibility to use either W3C DOM or IE 4 DOM depending on their availability, then you can use a capability-testing approach that first checks for the existence of a method or property to determine whether the browser has the capability you desire. For example −

if (document.getElementById) {

// If the W3C method exists, use it

}

else if (document.all) {

// If the all[] array exists, use it

}

else {

// Otherwise use the legacy DOM

}

## The onerror() Method

The onerror event handler was the first feature to facilitate error handling in JavaScript. The error event is fired on the window object whenever an exception occurs on the page.

The **onerror** event handler provides three pieces of information to identify the exact nature of the error −

Error message − The same message that the browser would display for the given error

URL − The file in which the error occurred

Line number− The line number in the given URL that caused the error.

window.onerror = function (msg, url, line) {

alert("Message : " + msg );

alert("url : " + url );

alert("Line number : " + line );

}

**Question: How to set a default parameter value for a JavaScript function?**

/\*\* Here email is parameter in which we have set the default value i.e email@domain.com \*\*/

function function1(name, email)

{

email = typeof email !== 'undefined' ? email : 'defaultemail@domain.com';

console.log('name='+name+', Email= '+email);

}

function1('john','myname@gmail.com');

function1('john');

**Queston: How to convert a string to lowercase?**

var str='This is testing String';

str = str.toLowerCase();

console.log(str);

**Question: How to modify the URL of page without reloading the page?**  
use pushState javascript function.   
**For Example:**

window.history.pushState('page2', 'This is page Title', '/newpage.php');

**Question: How to convert JSON Object to String?**

var myobject=['Web','Technology','Experts','Notes']

JSON.stringify(myobject);

**Question: How to convert JSON String to Object?**

var jsonData = '{"name":"web technology","year":2015}';

var myobject = JSON.parse(jsonData);

console.log(myobject);

**Question: How to check an variable is Object OR String OR Array?**  
Use below function to get Data type of javascript variable.

function checkDataType(someVar){

result ='String';

if(someVar instanceof Object){

result ='Object'

}

if($.isArray(someVar)){

result = 'Array';

}

return result;

}

var someVar= new Array("Saab", "Volvo", "BMW");

console.log(result);

**Question: Can i declare a variable as CONSTANT like in PHP?**  
**No,** I think cosntant not exist in javascript.   
But you can follow same type convention to declare constant.

var CONSTANT\_NAME = "constant value";

**Question: How to open URL in new tab in javascript?**  
use javascript, window.open function.

window.open('http://www.web-technology-experts-notes.in/','\_blank');

**Question: What is difference between undefined and object?**  
**undefined**means some variable's value is not defined yet.  
**object**means variables's value is defined that is either function, object OR array.  
  
**With use of below, you can easily determine whether it is object OR NULL.**

console.log(typeof(null)); // object

console.log(typeof(undefined)); // undefined

**Question: How to get current date in JavaScript?**

var today = new Date();

console.log(today);

**Question: How do I declare a namespace in JavaScript?**

var myNamespace = {

function1: function() { },

function2: function() { }

function3: function() { }

};

myNamespace.function3();

**Question: What is the best way to detect a mobile device in jQuery?** 

if( /Android|webOS|iPhone|iPad|iPod|BlackBerry|IEMobile|Opera Mini/i.test(navigator.userAgent) ) {

}

**Question: How to detect mobiles including ipad using navigator.useragent in javascript?**

if(navigator.userAgent.match(/Android/i) || navigator.userAgent.match(/webOS/i) || navigator.userAgent.match(/BlackBerry/i) || navigator.userAgent.match(/iPhone/i)){

console.log('Calling from Mobile');

}else{

console.log('Calling from Web');

}

**Question: How to detect mobiles including ipad using navigator.useragent in javascript?**

if(navigator.userAgent.match(/Android/i) || navigator.userAgent.match(/webOS/i) || navigator.userAgent.match(/BlackBerry/i) || navigator.userAgent.match(/iPhone/i)){

console.log('Calling from Mobile');

}else{

console.log('Calling from Web');

}

**<xsl:key>** tag element specifies a named name-value pair assigned to a specific element in an XML document. This key is used with the key() function in XPath expressions to access the assigned elements in a XML document.

<xsl:key

name = QName

match = Pattern

use = Expression

</xsl:key>

|  |  |
| --- | --- |
| name | Name of the key to be used. |
| match | Patterns used to identify a node that holds this key. |
| use | XPath expression to identified the value of the nodes of xml document. |