Table of Contents

[Language 2](#_Toc493185787)

[Language Statements 2](#_Toc493185788)

[Language Operators 2](#_Toc493185789)

[Language String 2](#_Toc493185790)

[Language Number 2](#_Toc493185791)

[Language Math 3](#_Toc493185792)

[Language Date 3](#_Toc493185793)

[Language Array 4](#_Toc493185794)

[Language Boolean 4](#_Toc493185795)

[Language RegExp 4](#_Toc493185796)

[Language Global 5](#_Toc493185797)

[Window Object 5](#_Toc493185798)

[DOM Window Properties 5](#_Toc493185799)

[Window Methods 7](#_Toc493185800)

[DOM navigator 8](#_Toc493185801)

[DOM screen 8](#_Toc493185802)

[DOM history 9](#_Toc493185803)

[DOM location 9](#_Toc493185804)

[DOM document 9](#_Toc493185805)

[CSS Selector Reference 10](#_Toc493185806)

[DOM Element 12](#_Toc493185807)

[DOM Attribute 14](#_Toc493185808)

[DOM Events 15](#_Toc493185809)

[Stop Event Propagation 20](#_Toc493185810)

[jQuery Selectors 20](#_Toc493185811)

[jQuery Misc 20](#_Toc493185812)

[CSS POSITIONING 21](#_Toc493185813)

[HOW TO DISPLAY DIVs SIDE BY SIDE 21](#_Toc493185814)

[HOW TO CREATE A TABLE USING UL AND LI TAGS 21](#_Toc493185815)

[SAMPLE JQUERY PLUGIN 21](#_Toc493185816)

[EXTEND OBJECTS WITH PROTOTYPE 24](#_Toc493185817)

[Inheritance thru prototype 25](#_Toc493185818)

[Convenient Inheritance 26](#_Toc493185819)

[MIXIN 27](#_Toc493185820)

[CLASSIC COPY 27](#_Toc493185821)

[EXTEND OBJECTS WITH JQUERY 28](#_Toc493185822)

[INHERITING PARTS OF VARIOUS CLASSES 28](#_Toc493185823)

[PRETTY PRINT 28](#_Toc493185824)

[TRAVERSE ALL DIVS AND HIGHLIGHT THEM WITH RED BORDER 29](#_Toc493185825)

[NODE.JS 29](#_Toc493185826)

[Installations 29](#_Toc493185827)

[NodeJs File IO 30](#_Toc493185828)

[http request 30](#_Toc493185829)

[request GET 30](#_Toc493185830)

[request POST 30](#_Toc493185831)

[uncompress gzip response 31](#_Toc493185832)

[Express and serve html file 31](#_Toc493185833)

[Http module GET/POST 31](#_Toc493185834)

# Language

## Language Statements

Break; continue; debugger, do {} while (true); for (var s in ss) {}; function(){}; if (){} else {} else if () {}; return; switch(s) {case: {} break; default: {}}; throw "Too big"; throw 500; try {} catch(err){} finally {}; var x = 0; while(true){}

## Language Operators

+, -, \*, /, %, ++, --

=, +=, -=, \*=, /=, %=

==, ===, !=, !==, >, <, >=, <=

&&, ||, !

&, |, ~, ^, >>, <<

## Language String

String.**constructor**: function String() { [native code] }

.**length**, .**prototype**

“hell”.charAt(0): h

“hell”.charCodeAt(0): 104

“hello”.concat(“ world”): hello world

String.fromCharCode(104): h

"Hello world, welcome to the universe.".indexOf("welcome"): 13

"Hello planet earth, you are a great planet.".lastIndexOf("planet"): 36

"ab".localeCompare("ab"): 0

"(714)123-4567".**match**(/\(\d{3}\)\d{3}-\d{4}/): (714)123-4567

"1/2/2014".**replace**(/(\d{1,2})\/(\d{1,2})\/(\d{2,4})/, "$3-$1-$2"): 2014-1-2

out="Visit W3Schools!".**search**("W3Schools"): 6

"Hello world!".**slice**(1,5): ello //2nd arg exclusive

"How are you doing today?".**split**(" "): How,are,you,doing,today?

"Hello world!".**substr**(1, 4): ello //inclusive of second argument

"Hello world!".**substring**(1, 4): ell //exclusive of second argument, same as slice

"Hello World!".toLocaleLowerCase(): hello world!

"Hello World!".toLocaleUpperCase(): HELLO WORLD!

"Hello World!".toLowerCase(): hello world!

"Hello World!".toUpperCase(): HELLO WORLD!

"Hello World!".toString(): Hello World!

“xxx “.trim(): “xxx”

"Hello World!".valueOf(): Hello World!

## Language Number

134.5.**constructor**: function Number() { [native code] }

Number.MAX\_VALUE: 1.7976931348623157e+308

Number.**MIN\_VALUE**: 5e-324

Number.**NEGATIVE\_INFINITY** : -Infinity

Number.**POSITIVE\_INFINITY** : Infinity

Number.**NaN**: NaN

Number.**prototype**

5.56789.**toExponential**():5.56789e+0

5.56789.**toFixed**(2): 5.57

13.3714.**toPrecision**(1): 1e+1

13.3714.**toPrecision**(2): 13

1e+1.**toString**(): 10

1e+1.**valueOf**(): 10

## Language Math

|  |  |  |
| --- | --- | --- |
| Math.**E** | : | 2.718281828 |
| Math.**LN2** | : | 0.693147181 |
| Math.LN10 | : | 2.302585093 |
| Math.LOG2E | : | 1.442695041 |
| Math.LOG10E | : | 0.434294482 |
| Math.PI | : | 3.141592654 |
| Math.SQRT1\_2 | : | 0.707106781 |
| Math.SQRT2 | : | 1.414213562 |

abs(x) Returns the absolute value of x

acos(x) Returns the arccosine of x, in radians

asin(x) Returns the arcsine of x, in radians

atan(x) Returns the arctangent of x as a numeric value between -PI/2 and PI/2 radians

atan2(y,x) Returns the arctangent of the quotient of its arguments

ceil(x) Returns x, rounded upwards to the nearest integer

cos(x) Returns the cosine of x (x is in radians)

exp(x) Returns the value of Ex

floor(x) Returns x, rounded downwards to the nearest integer

log(x) Returns the natural logarithm (base E) of x

max(x,y,z,...,n) Returns the number with the highest value

min(x,y,z,...,n) Returns the number with the lowest value

pow(x,y) Returns the value of x to the power of y

random() Returns a random number between 0 and 1

round(x) Rounds x to the nearest integer

sin(x) Returns the sine of x (x is in radians)

sqrt(x) Returns the square root of x

tan(x) Returns the tangent of an angle

## Language Date

constructor, prototype

Local:12/11/2014 14:56:14.10  
UTC:12/11/2014 22:56:14.10

getDate() Returns the day of the month (from 1-31)

getDay() Returns the day of the week (from 0-6)

getFullYear() Returns the year (four digits)

getHours() Returns the hour (from 0-23)

getMilliseconds() Returns the milliseconds (from 0-999)

getMinutes() Returns the minutes (from 0-59)

getMonth() Returns the month (from 0-11)

getSeconds() Returns the seconds (from 0-59)

getTime() Returns the number of milliseconds since midnight Jan 1, 1970

getTimezoneOffset() Returns the time difference between UTC time and local time, in minutes

getUTCDate() Returns the day of the month, according to universal time (from 1-31)

getUTCDay() Returns the day of the week, according to universal time (from 0-6)

getUTCFullYear() Returns the year, according to universal time (four digits)

getUTCHours() Returns the hour, according to universal time (from 0-23)

getUTCMilliseconds() Returns the milliseconds, according to universal time (from 0-999)

getUTCMinutes() Returns the minutes, according to universal time (from 0-59)

getUTCMonth() Returns the month, according to universal time (from 0-11)

getUTCSeconds() Returns the seconds, according to universal time (from 0-59)

getYear() Deprecated. Use the getFullYear() method instead

parse("March 21, 2012") Parses a date string and returns the number of milliseconds since January 1, 1970

setDate(15) Sets the day of the month of a date object

setFullYear(2014) Sets the year (four digits) of a date object

setHours(15) Sets the hour of a date object

setMilliseconds(192) Sets the milliseconds of a date object

setMinutes(17) Set the minutes of a date object

setMonth(4) Sets the month of a date object, 0 based.

setSeconds(35) Sets the seconds of a date object

setTime(1332403882588) Sets a date to a specified number of milliseconds after/before January 1, 1970

setUTCDate(15) Sets the day of the month of a date object, according to universal time

setUTCFullYear(2014) Sets the year of a date object, according to universal time (four digits)

setUTCHours(15) Sets the hour of a date object, according to universal time

setUTCMilliseconds(192) Sets the milliseconds of a date object, according to universal time

setUTCMinutes(17) Set the minutes of a date object, according to universal time

setUTCMonth(4) Sets the month of a date object, according to universal time

setUTCSeconds(35) Set the seconds of a date object, according to universal time

setYear() Deprecated. Use the setFullYear() method instead

toDateString() “Fri Dec 12 2014”

toGMTString() Deprecated. Use the toUTCString() method instead

toISOString() “2014-12-12T23:07:49.128Z”

toJSON() “2014-12-12T23:08:26.303Z”

toLocaleDateString() “12/12/2014”

toLocaleTimeString() “11:44:46 PM”

toLocaleString() “12/12/2014, 11:45:03 PM”

toString() “Fri Dec 12 2014 23:45:58 GMT+0000 (Coordinated Universal Time)”

toTimeString() “23:47:23 GMT+0000 (Coordinated Universal Time)”

toUTCString() “Fri, 12 Dec 2014 23:47:48 GMT”

UTC() 1333065600000

valueOf() 1418428100923

## Language Array

**constructor**, **prototype**, **length**

["Cecilie", "Lone"].**concat**(["Emil", "Tobias", "Linus"]): Cecilie,Lone,Emil,Tobias,Linus

["Banana", "Orange", "Apple", "Mango"].**indexOf**("Apple"): 2

["Banana", "Orange", "Apple", "Mango"].**join**():Banana,Orange,Apple,Mango

["Banana", "Apple", "Apple", "Mango"].**lastIndexOf**("Apple"): 2

["Banana", "Orange", "Apple", "Mango"].**pop**(): “Mango” //only first 3 remains

["Banana", "Orange", "Apple", "Mango"].**push**("More Fruit"): 5

["Banana", "Orange", "Apple", "Mango"].**reverse**().join():Mango,Apple,Orange,Banana

["Banana", "Orange", "Apple", "Mango"].**shift**(): Banana //only last 3 elements remain, “shift” to left

["Banana", "Orange", "Lemon", "Apple", "Mango"].**slice**(1,3).join():Orange,Lemon //2nd arg exclusive

["Banana", "Orange", "Lemon", "Apple", "Mango"].**sort**().join(): Apple,Banana,Lemon,Mango,Orange

["Banana", "Orange", "Apple", "Mango"].**splice**(2,1,"Lemon","Kiwi"): Apple // remaining: Banana,Orange,Lemon,Kiwi,Mango

["Banana", "Orange", "Apple", "Mango"].**toString**():Banana,Orange,Apple,Mango

["Banana", "Orange", "Apple", "Mango"].**unshift**("Lemon","Pineapple"): 6 //Lemon,Pineapple,Banana,Orange,Apple,Mango

["Banana", "Orange", "Apple", "Mango"].**valueOf**():Banana,Orange,Apple,Mango

## Language Boolean

**constructor**, prototype

true.**valueOf**(): true

false.**toString**(): false

## Language RegExp

|  |  |
| --- | --- |
| **Modifier** | **Description** |
| [i](http://www.w3schools.com/jsref/jsref_regexp_i.asp) | Perform case-insensitive matching |
| [g](http://www.w3schools.com/jsref/jsref_regexp_g.asp) | Perform a global match (find all matches rather than stopping after the first match) |
| [m](http://www.w3schools.com/jsref/jsref_regexp_m.asp) | Perform multiline matching |

|  |  |
| --- | --- |
| **Expression** | **Description** |
| [[abc]](http://www.w3schools.com/jsref/jsref_regexp_charset.asp) | Find any character between the brackets |
| [[^abc]](http://www.w3schools.com/jsref/jsref_regexp_charset_not.asp) | Find any character NOT between the brackets |
| [[0-9]](http://www.w3schools.com/jsref/jsref_regexp_0-9.asp) | Find any digit between the brackets |
| [[^0-9]](http://www.w3schools.com/jsref/jsref_regexp_not_0-9.asp) | Find any digit NOT between the brackets |
| [(x|y)](http://www.w3schools.com/jsref/jsref_regexp_xy.asp) | Find any of the alternatives specified |

|  |  |
| --- | --- |
| **Metacharacter** | **Description** |
| [.](http://www.w3schools.com/jsref/jsref_regexp_dot.asp) | Find a single character, except newline or line terminator |
| [\w](http://www.w3schools.com/jsref/jsref_regexp_wordchar.asp) | Find a word character |
| [\W](http://www.w3schools.com/jsref/jsref_regexp_wordchar_non.asp) | Find a non-word character |
| [\d](http://www.w3schools.com/jsref/jsref_regexp_digit.asp) | Find a digit |
| [\D](http://www.w3schools.com/jsref/jsref_regexp_digit_non.asp) | Find a non-digit character |
| [\s](http://www.w3schools.com/jsref/jsref_regexp_whitespace.asp) | Find a whitespace character |
| [\S](http://www.w3schools.com/jsref/jsref_regexp_whitespace_non.asp) | Find a non-whitespace character |
| [\b](http://www.w3schools.com/jsref/jsref_regexp_begin.asp) | Find a match at the beginning/end of a word |
| [\B](http://www.w3schools.com/jsref/jsref_regexp_begin_not.asp) | Find a match not at the beginning/end of a word |
| [\0](http://www.w3schools.com/jsref/jsref_regexp_nul.asp) | Find a NUL character |
| [\n](http://www.w3schools.com/jsref/jsref_regexp_newline.asp) | Find a new line character |
| [\f](http://www.w3schools.com/jsref/jsref_regexp_formfeed.asp) | Find a form feed character |
| [\r](http://www.w3schools.com/jsref/jsref_regexp_carriagereturn.asp) | Find a carriage return character |
| [\t](http://www.w3schools.com/jsref/jsref_regexp_tab.asp) | Find a tab character |
| [\v](http://www.w3schools.com/jsref/jsref_regexp_vtab.asp) | Find a vertical tab character |
| [\](http://www.w3schools.com/jsref/jsref_regexp_octal.asp)nnn | Find the character specified by an octal number nnn |
| [\xdd](http://www.w3schools.com/jsref/jsref_regexp_hex.asp) | Find the character specified by a hexadecimal number dd |
| [\uxxxx](http://www.w3schools.com/jsref/jsref_regexp_unicode_hex.asp) | Find the Unicode character specified by a hexadecimal number xxxx |

|  |  |
| --- | --- |
| **Quantifier** | **Description** |
| [n+](http://www.w3schools.com/jsref/jsref_regexp_onemore.asp) | Matches any string that contains at least one *n* |
| [n\*](http://www.w3schools.com/jsref/jsref_regexp_zeromore.asp) | Matches any string that contains zero or more occurrences of *n* |
| [n?](http://www.w3schools.com/jsref/jsref_regexp_zeroone.asp) | Matches any string that contains zero or one occurrences of *n* |
| [n{X}](http://www.w3schools.com/jsref/jsref_regexp_nx.asp) | Matches any string that contains a sequence of *X* *n*'s |
| [n{X,Y}](http://www.w3schools.com/jsref/jsref_regexp_nxy.asp) | Matches any string that contains a sequence of X to Y *n*'s |
| [n{X,}](http://www.w3schools.com/jsref/jsref_regexp_nxcomma.asp) | Matches any string that contains a sequence of at least X *n*'s |
| [n$](http://www.w3schools.com/jsref/jsref_regexp_ndollar.asp) | Matches any string with *n* at the end of it |
| [^n](http://www.w3schools.com/jsref/jsref_regexp_ncaret.asp) | Matches any string with *n* at the beginning of it |
| [?=n](http://www.w3schools.com/jsref/jsref_regexp_nfollow.asp) | Matches any string that is followed by a specific string *n* |
| [?!n](http://www.w3schools.com/jsref/jsref_regexp_nfollow_not.asp) | Matches any string that is not followed by a specific string *n* |

constructor

var r = /\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}/g; document.write(r.**global**); //true, ip address pattern

var r = /\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}/g; document.write(r.**ignoreCase**); //false

var str = "The rain in Spain stays mainly in the plain"; var patt1 = /ain/g; while (patt1.test(str)==true) { document.write(patt1.lastIndex+','); } //8,17,28,43,

var r = /\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}/g; document.write(r.**multiline**); //false

var r = /\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}/g; document.write(r.**source**); //\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}

/\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}/g.**test**('127.0.0.1') //true

/\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}/g.**exec**('127.0.0.1') //127.0.0.1

/\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}/g.**toString**() /\* /\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}/g \*/

## Language Global

document.write(1.7976931348623157E+10308 + ", " + -1.7976931348623157E+10308); //**Infinity**, -**Infinity**

document.write(**isNaN**(NaN)); //true

document.write(typeof **undefined** === 'undefined'); //undefined

document.write(**encodeURI**("my test.asp?name=ståle&car=saab")); //my%20test.asp?name=st%C3%83%C2%A5le&car=saab

document.write(**decodeURI**(encodeURI("my test.asp?name=ståle&car=saab"))); //my test.asp?name=stÃ¥le&car=saab

**encodeURIComponent**()/**decodeURIComponent**() //http parameters

**eval**('document.write("test")'); //test

document.write(**isFinite**(1.7976931348623157E+10308)); //false

document.write(**Number**(1.7976931348623157E+10308)); //Infinity

**parseFloat**("40 years"): 40

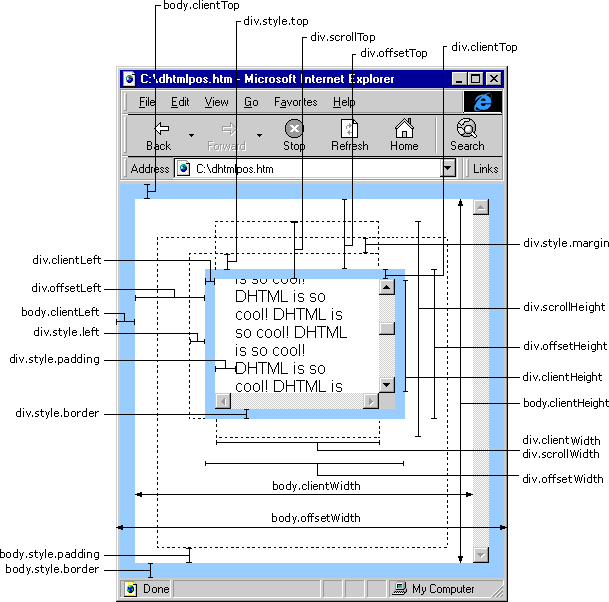
**parseInt**("0x10"): 16

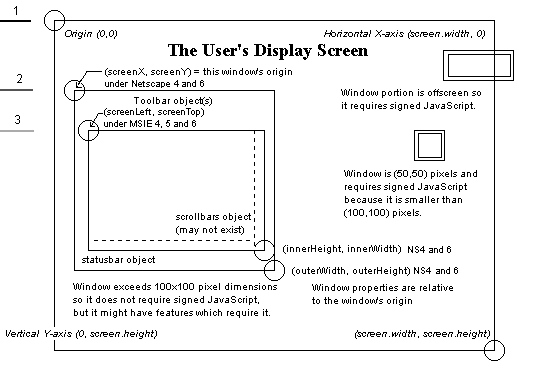
**String**(new Date()):Fri Dec 12 2014 17:40:48 GMT-0800 (Pacific Standard Time)

# Window Object

## DOM Window Properties

|  |  |
| --- | --- |
| **Property** | **Description** |
| [closed](http://www.w3schools.com/jsref/prop_win_closed.asp) | Returns a Boolean value indicating whether a window has been closed or not |
| [defaultStatus](http://www.w3schools.com/jsref/prop_win_defaultstatus.asp) | Sets or returns the default text in the statusbar of a window |
| [document](http://www.w3schools.com/jsref/dom_obj_document.asp) | [Returns the Document object for the window (See Document object)](http://www.w3schools.com/jsref/dom_obj_document.asp) |
| [frameElement](http://www.w3schools.com/jsref/prop_win_frameElement.asp) | Returns the <iframe> element in which the current window is sitting in |
| [frames](http://www.w3schools.com/jsref/prop_win_frames.asp) | Returns all <iframe> elements in the current window |
| [history](http://www.w3schools.com/jsref/obj_history.asp) | [Returns the History object for the window (See History object)](http://www.w3schools.com/jsref/obj_history.asp) |
| [innerHeight](http://www.w3schools.com/jsref/prop_win_innerheight.asp) | Returns the inner height of a window's content area |
| [innerWidth](http://www.w3schools.com/jsref/prop_win_innerheight.asp) | Returns the inner width of a window's content area |
| [length](http://www.w3schools.com/jsref/prop_win_length.asp) | Returns the number of <iframe> elements in the current window |
| [location](http://www.w3schools.com/jsref/obj_location.asp) | [Returns the Location object for the window (See Location object)](http://www.w3schools.com/jsref/obj_location.asp) |
| [name](http://www.w3schools.com/jsref/prop_win_name.asp) | Sets or returns the name of a window |
| [navigator](http://www.w3schools.com/jsref/obj_navigator.asp) | [Returns the Navigator object for the window (See Navigator object)](http://www.w3schools.com/jsref/obj_navigator.asp) |
| [opener](http://www.w3schools.com/jsref/prop_win_opener.asp) | Returns a reference to the window that created the window |
| [outerHeight](http://www.w3schools.com/jsref/prop_win_outerheight.asp) | Returns the outer height of a window, including toolbars/scrollbars |
| [outerWidth](http://www.w3schools.com/jsref/prop_win_outerheight.asp) | Returns the outer width of a window, including toolbars/scrollbars |
| [pageXOffset](http://www.w3schools.com/jsref/prop_win_pagexoffset.asp) | Returns the pixels the current document has been scrolled (horizontally) from the upper left corner of the window |
| [pageYOffset](http://www.w3schools.com/jsref/prop_win_pagexoffset.asp) | Returns the pixels the current document has been scrolled (vertically) from the upper left corner of the window |
| [parent](http://www.w3schools.com/jsref/prop_win_parent.asp) | Returns the parent window of the current window |
| [screen](http://www.w3schools.com/jsref/obj_screen.asp) | [Returns the Screen object for the window (See Screen object)](http://www.w3schools.com/jsref/obj_screen.asp) |
| [screenLeft](http://www.w3schools.com/jsref/prop_win_screenleft.asp) | Returns the horizontal coordinate of the window relative to the screen |
| [screenTop](http://www.w3schools.com/jsref/prop_win_screenleft.asp) | Returns the vertical coordinate of the window relative to the screen |
| [screenX](http://www.w3schools.com/jsref/prop_win_screenx.asp) | Returns the horizontal coordinate of the window relative to the screen, including toolbar and status bar |
| [screenY](http://www.w3schools.com/jsref/prop_win_screenx.asp) | Returns the vertical coordinate of the window relative to the screen, including toolbar and status bar |
| scrollX | [An alias of pageXOffset](http://www.w3schools.com/jsref/prop_win_pagexoffset.asp) |
| scrollY | [An alias of pageYOffset](http://www.w3schools.com/jsref/prop_win_pagexoffset.asp) |
| [self](http://www.w3schools.com/jsref/prop_win_self.asp) | Returns the current window |
| [status](http://www.w3schools.com/jsref/prop_win_status.asp) | Sets or returns the text in the statusbar of a window |
| [top](http://www.w3schools.com/jsref/prop_win_top.asp) | Returns the topmost browser window |





## Window Methods

|  |  |
| --- | --- |
| **Method** | **Description** |
| [alert(‘hello’)](http://www.w3schools.com/jsref/met_win_alert.asp) | Displays an alert box with a message and an OK button |
| [atob(‘SGVsbG8gV29ybGQh’)](http://www.w3schools.com/jsref/met_win_atob.asp) | Decodes a base-64 encoded string, ‘Hello world!’ (think of b as byte or string) |
| [blur()](http://www.w3schools.com/jsref/met_win_blur.asp) | Removes focus from the current window |
| [btoa(‘Hello world!’)](http://www.w3schools.com/jsref/met_win_btoa.asp) | Encodes a string in base-64, SGVsbG8gV29ybGQh (think of b as byte or string) |
| [clearInterval()](http://www.w3schools.com/jsref/met_win_clearinterval.asp) | var myVar = setInterval(function(){ window.status = 'xyz'; }, 1000); clearInterval(myVar); |
| [clearTimeout()](http://www.w3schools.com/jsref/met_win_cleartimeout.asp) | var myVar = setTimeout(function(){ window.status = 'xyz'; }, 1000); clearTimeout(myVar); |
| [close()](http://www.w3schools.com/jsref/met_win_close.asp) | Closes the current window |
| [confirm()](http://www.w3schools.com/jsref/met_win_confirm.asp) | Var tmp = window.confirm(‘xyz’); //true(ok), false (cancel) |
| [createPopup()](http://www.w3schools.com/jsref/met_win_createpopup.asp) | //IE    var p = window.createPopup()   var pbody = p.document.body   pbody.style.backgroundColor = "lime"   pbody.style.border = "solid black 1px"   pbody.innerHTML = "This is a pop-up! Click outside to close."   p.show(150,150,200,50,document.body) |
| [focus()](http://www.w3schools.com/jsref/met_win_focus.asp) | Sets focus to the current window |
| [moveBy()](http://www.w3schools.com/jsref/met_win_moveby.asp) | //IE  myWindow = window.open('', '', 'width=200, height=100');  myWindow.document.write("<p>This is 'myWindow'</p>");  myWindow.moveBy(250, 250); //relative to screen |
| [moveTo()](http://www.w3schools.com/jsref/met_win_moveto.asp) | //IE  myWindow = window.open('', '', 'width=200, height=100');  myWindow.document.write("<p>This is 'myWindow'</p>");  myWindow.moveTo(500, 100); //relative to screen |
| [open()](http://www.w3schools.com/jsref/met_win_open.asp) | window.open("http://www.w3schools.com"); |
| [print()](http://www.w3schools.com/jsref/met_win_print.asp) | Prints the content of the current window |
| [prompt()](http://www.w3schools.com/jsref/met_win_prompt.asp) | var person = prompt("Please enter your name", "Harry Potter");  window.status = person; |
| [resizeBy()](http://www.w3schools.com/jsref/met_win_resizeby.asp) | var myWindow = window.open("", "", "width=100, height=100");  myWindow.resizeBy(250, 250); |
| [resizeTo()](http://www.w3schools.com/jsref/met_win_resizeto.asp) | var myWindow = window.open("", "", "width=100, height=100");  myWindow.resizeTo(250, 250); |
| scroll() | [Deprecated. This method has been replaced by the scrollTo() method.](http://www.w3schools.com/jsref/met_win_scrollto.asp) |
| [scrollBy()](http://www.w3schools.com/jsref/met_win_scrollby.asp) | window.scrollBy(100, 0); |
| [scrollTo()](http://www.w3schools.com/jsref/met_win_scrollto.asp) | window.scrollTo(500, 0); |
| [setInterval()](http://www.w3schools.com/jsref/met_win_setinterval.asp) | var myVar = setInterval(function(){ window.status = 'xyz'; }, 1000); clearInterval(myVar); |
| [setTimeout()](http://www.w3schools.com/jsref/met_win_settimeout.asp) | var myVar = setTimeout(function(){ window.status = 'xyz'; }, 1000); clearTimeout(myVar); |
| [stop()](http://www.w3schools.com/jsref/met_win_stop.asp) | Stops the window from loading (non-IE) |

# DOM navigator

|  |  |
| --- | --- |
| [appCodeName](http://www.w3schools.com/jsref/prop_nav_appcodename.asp) | Mozilla |
| [appName](http://www.w3schools.com/jsref/prop_nav_appname.asp) | Netscape |
| [appVersion](http://www.w3schools.com/jsref/prop_nav_appversion.asp) | 5.0 (Windows) |
| [cookieEnabled](http://www.w3schools.com/jsref/prop_nav_cookieenabled.asp) | TRUE |
| [geolocation](http://www.w3schools.com/jsref/prop_nav_geolocation.asp) | [object Geolocation] |
| [language](http://www.w3schools.com/jsref/prop_nav_language.asp) | en-US |
| [onLine](http://www.w3schools.com/jsref/prop_nav_online.asp) | TRUE |
| [platform](http://www.w3schools.com/jsref/prop_nav_platform.asp) | Win32 |
| [product](http://www.w3schools.com/jsref/prop_nav_product.asp) | Gecko |
| [userAgent](http://www.w3schools.com/jsref/prop_nav_useragent.asp) | Mozilla/5.0 (Windows NT 6.1; rv:31.0) Gecko/20100101 Firefox/31.0 |
| javaEnabled() | TRUE |

# DOM screen

|  |  |
| --- | --- |
| [availHeight](http://www.w3schools.com/jsref/prop_screen_availheight.asp) | 760 (excluding windows taskbar at bottom) |
| [availWidth](http://www.w3schools.com/jsref/prop_screen_availwidth.asp) | 1280 |
| [colorDepth](http://www.w3schools.com/jsref/prop_screen_colordepth.asp) | 24 |
| [height](http://www.w3schools.com/jsref/prop_screen_height.asp) | 800 |
| [pixelDepth](http://www.w3schools.com/jsref/prop_screen_pixeldepth.asp) | 24 |
| [width](http://www.w3schools.com/jsref/prop_screen_width.asp) | 1280 |

# DOM history

|  |  |
| --- | --- |
| **History Object Properties** |  |
|  |  |
| **Property** | **Description** |
| [length](http://www.w3schools.com/jsref/prop_his_length.asp) | Returns the number of URLs in the history list |
|  |  |
| **History Object Methods** |  |
|  |  |
| **Method** | **Description** |
| [back()](http://www.w3schools.com/jsref/met_his_back.asp) | Loads the previous URL in the history list |
| [forward()](http://www.w3schools.com/jsref/met_his_forward.asp) | Loads the next URL in the history list |
| [go()](http://www.w3schools.com/jsref/met_his_go.asp) | Loads a specific URL from the history list |

# DOM location

|  |  |
| --- | --- |
| [hash](http://www.w3schools.com/jsref/prop_loc_hash.asp) | Sets or returns the anchor part (#) of a URL |
| [host](http://www.w3schools.com/jsref/prop_loc_host.asp) | Hostname + port number |
| [hostname](http://www.w3schools.com/jsref/prop_loc_hostname.asp) | hostname |
| [href](http://www.w3schools.com/jsref/prop_loc_href.asp) | file:///C:/TEMP/test.html |
| [origin](http://www.w3schools.com/jsref/prop_loc_origin.asp) | Returns the protocol, hostname and port number of a URL (href without path) |
| [pathname](http://www.w3schools.com/jsref/prop_loc_pathname.asp) | /C:/TEMP/test.html |
| [port](http://www.w3schools.com/jsref/prop_loc_port.asp) |  |
| [protocol](http://www.w3schools.com/jsref/prop_loc_protocol.asp) | file: |
| [search](http://www.w3schools.com/jsref/prop_loc_search.asp) | Query string |

Assign(‘http://www.google.com’), reload(); replace(‘http://www.google.com’); /\*replace current doc, current doc will not be in history \*/

# DOM document

|  |  |
| --- | --- |
| [document.addEventListener()](http://www.w3schools.com/jsref/met_document_addeventlistener.asp) | document.addEventListener("click", function(){     document.getElementById("demo").innerHTML = "Hello World"; }); |
| [document.adoptNode()](http://www.w3schools.com/jsref/met_document_adoptnode.asp) | var frame = document.getElementsByTagName("IFRAME")[0]  var h = frame.contentWindow.document.getElementsByTagName("H1")[0];  var x = document.adoptNode(h);  document.body.appendChild(x); |
| [document.anchors](http://www.w3schools.com/jsref/coll_doc_anchors.asp) | Returns a collection of all <a> elements in the document that have a name attribute |
| [document.applets](http://www.w3schools.com/jsref/coll_doc_applets.asp) | Returns a collection of all <applet> elements in the document |
| [document.baseURI](http://www.w3schools.com/jsref/prop_doc_baseuri.asp) | Returns the absolute base URI of a document (non-IE, equivalent to origin?) |
| [document.body](http://www.w3schools.com/jsref/prop_doc_body.asp) | Sets or returns the document's body (the <body> element) |
| [document.close()](http://www.w3schools.com/jsref/met_doc_close.asp) | document.open(); document.write("<h1>Hello World</h1>"); document.close(); |
| [document.cookie](http://www.w3schools.com/jsref/prop_doc_cookie.asp) | Returns all name/value pairs of cookies in the document |
| [document.createAttribute()](http://www.w3schools.com/jsref/met_document_createattribute.asp) | var h1 = document.getElementsByTagName("H1")[0];  var att = document.createAttribute("class");  att.value = "democlass";  h1.**setAttributeNode**(att); |
| [document.createComment()](http://www.w3schools.com/jsref/met_document_createcomment.asp) | var c = document.createComment("My personal comments"); document.body.**appendChild**(c); |
| [document.createDocumentFragment()](http://www.w3schools.com/jsref/met_document_createdocumentfragment.asp) | var d = document.createDocumentFragment(); d.appendChild(document.getElementsByTagName("LI")[0]); d.childNodes[0].childNodes[0].nodeValue = "Milk"; document.getElementsByTagName("UL")[0].**appendChild**(d); |
| [document.createElement()](http://www.w3schools.com/jsref/met_document_createelement.asp) | var btn = document.createElement("BUTTON"); |
| [document.createTextNode()](http://www.w3schools.com/jsref/met_document_createtextnode.asp) | var t = document.createTextNode("Hello World"); |
| [document.doctype](http://www.w3schools.com/jsref/prop_document_doctype.asp) | var x = document.doctype.name; //html |
| [document.documentElement](http://www.w3schools.com/jsref/prop_document_documentelement.asp) | var x = document.documentElement.nodeName; //HTML node |
| [document.documentMode](http://www.w3schools.com/jsref/prop_doc_documentmode.asp) | var x = document.documentMode; //11 – IE version |
| [document.documentURI](http://www.w3schools.com/jsref/prop_document_documenturi.asp) | Sets or returns the location of the document (any document type) |
| [document.domain](http://www.w3schools.com/jsref/prop_doc_domain.asp) | Returns the domain name of the server that loaded the document, www.pyfia.com |
| document.domConfig | Obsolete. Returns the DOM configuration of the document |
| [document.embeds](http://www.w3schools.com/jsref/coll_doc_embeds.asp) | Returns a collection of all <embed> elements the document |
| [document.forms](http://www.w3schools.com/jsref/coll_doc_forms.asp) | Returns a collection of all <form> elements in the document |
| [document.getElementById()](http://www.w3schools.com/jsref/met_document_getelementbyid.asp) | Returns the element that has the ID attribute with the specified value |
| [document.getElementsByClassName()](http://www.w3schools.com/jsref/met_document_getelementsbyclassname.asp) | Returns a NodeList containing all elements with the specified class name |
| [document.getElementsByName()](http://www.w3schools.com/jsref/met_doc_getelementsbyname.asp) | Returns a NodeList containing all elements with a specified name |
| [document.getElementsByTagName()](http://www.w3schools.com/jsref/met_document_getelementsbytagname.asp) | Returns a NodeList containing all elements with the specified tag name |
| [document.head](http://www.w3schools.com/jsref/prop_doc_head.asp) | Returns the <head> element of the document |
| [document.images](http://www.w3schools.com/jsref/coll_doc_images.asp) | Returns a collection of all <img> elements in the document |
| [document.implementation](http://www.w3schools.com/jsref/prop_document_implementation.asp) | var imp = document.implementation;  document.getElementById("demo").innerHTML = imp.hasFeature("HTML", "1.0"); |
| [document.importNode()](http://www.w3schools.com/jsref/met_document_importnode.asp) | var frame = document.getElementsByTagName("IFRAME")[0]  var h = frame.contentWindow.document.getElementsByTagName("H1")[0];  var x = document.importNode(h, true);  document.body.appendChild(x); |
| [document.inputEncoding](http://www.w3schools.com/jsref/prop_document_inputencoding.asp) | Returns the encoding, character set, used for the document, ISO-8859-1 |
| [document.lastModified](http://www.w3schools.com/jsref/prop_doc_lastmodified.asp) | Returns the date and time the document was last modified, 11/13/2014 10:05:21 |
| [document.links](http://www.w3schools.com/jsref/coll_doc_links.asp) | Returns a collection of all <a> and <area> elements in the document that have a href attribute |
| [document.normalize()](http://www.w3schools.com/jsref/met_document_normalize.asp) | Removes empty Text nodes, and joins adjacent text nodes |
| [document.normalizeDocument()](http://www.w3schools.com/jsref/met_document_normalizedocument.asp) | Removes empty Text nodes, and joins adjacent text nodes (not supported by any browser). |
| [document.open()](http://www.w3schools.com/jsref/met_doc_open.asp) | Opens an HTML output stream to collect output from document.write() |
| [document.querySelector()](http://www.w3schools.com/jsref/met_document_queryselector.asp) | Returns the first element that matches a specified CSS selector(s) in the document, document.querySelector(".example"); |
| [document.querySelectorAll()](http://www.w3schools.com/jsref/met_document_queryselectorall.asp) | Returns a static NodeList containing all elements that matches a specified CSS selector(s) in the document |
| [document.readyState](http://www.w3schools.com/jsref/prop_doc_readystate.asp) | Returns the (loading) status of the document, complete |
| [document.referrer](http://www.w3schools.com/jsref/prop_doc_referrer.asp) | Returns the URL of the document that loaded the current document |
| [document.removeEventListener()](http://www.w3schools.com/jsref/met_document_removeeventlistener.asp) | [Removes an event handler from the document (that has been attached with the addEventListener() method)](http://www.w3schools.com/jsref/met_document_addeventlistener.asp) |
| [document.renameNode()](http://www.w3schools.com/jsref/met_document_renamenode.asp) | var p = document.getElementsByTagName("P")[0]; document.renameNode(p, null, "h1"); |
| [document.scripts](http://www.w3schools.com/jsref/coll_doc_scripts.asp) | Returns a collection of <script> elements in the document |
| [document.strictErrorChecking](http://www.w3schools.com/jsref/prop_document_stricterrorchecking.asp) | Sets or returns whether error-checking is enforced or not, true/false |
| [document.title](http://www.w3schools.com/jsref/prop_doc_title.asp) | Sets or returns the title of the document |
| [document.URL](http://www.w3schools.com/jsref/prop_doc_url.asp) | Returns the full URL of the HTML document, html docs only |
| [document.write()](http://www.w3schools.com/jsref/met_doc_write.asp) | Writes HTML expressions or JavaScript code to a document |
| [document.writeln()](http://www.w3schools.com/jsref/met_doc_writeln.asp) | Same as write(), but adds a newline character after each statement //IE |

# CSS Selector Reference

|  |  |  |  |
| --- | --- | --- | --- |
| **Selector** | **Example** | **Example description** | **CSS** |
| [.class](http://www.w3schools.com/cssref/sel_class.asp) | .intro | Selects all elements with class="intro" | 1 |
| [#id](http://www.w3schools.com/cssref/sel_id.asp) | #firstname | Selects the element with id="firstname" | 1 |
| [\*](http://www.w3schools.com/cssref/sel_all.asp) | \* | Selects all elements | 2 |
| [element](http://www.w3schools.com/cssref/sel_element.asp) | p | Selects all <p> elements | 1 |
| [element,element](http://www.w3schools.com/cssref/sel_element_comma.asp) | div, p | Selects all <div> elements and all <p> elements | 1 |
| [element element](http://www.w3schools.com/cssref/sel_element_element.asp) | div p | Selects all <p> elements inside <div> elements | 1 |
| [element>element](http://www.w3schools.com/cssref/sel_element_gt.asp) | div > p | Selects all <p> elements where the parent is a <div> element | 2 |
| [element+element](http://www.w3schools.com/cssref/sel_element_pluss.asp) | div + p | Selects all <p> elements that are placed immediately after <div> elements | 2 |
| [element1~element2](http://www.w3schools.com/cssref/sel_gen_sibling.asp) | p ~ ul | Selects every <ul> element that are preceded by a <p> element | 3 |
| [[attribute]](http://www.w3schools.com/cssref/sel_attribute.asp) | [target] | Selects all elements with a target attribute | 2 |
| [[attribute=value]](http://www.w3schools.com/cssref/sel_attribute_value.asp) | [target**=**\_blank] | Selects all elements with target**=**"\_blank" | 2 |
| [[attribute~=value]](http://www.w3schools.com/cssref/sel_attribute_value_contains.asp) | [title**~=**flower] | Selects all elements with a title attribute **containing** the word "flower" | 2 |
| [[attribute|=value]](http://www.w3schools.com/cssref/sel_attribute_value_lang.asp) | [lang**|=**en] | Selects all elements with a lang attribute value starting with "en" | 2 |
| [[attribute^=value]](http://www.w3schools.com/cssref/sel_attr_begin.asp) | a[href**^=**"https"] | Selects every <a> element whose href attribute value begins with "https" | 3 |
| [[attribute$=value]](http://www.w3schools.com/cssref/sel_attr_end.asp) | a[href**$=**".pdf"] | Selects every <a> element whose href attribute value **ends** with ".pdf" | 3 |
| [[attribute\*=value]](http://www.w3schools.com/cssref/sel_attr_contain.asp) | a[href**\*=**"w3schools"] | Selects every <a> element whose href attribute value **contains** the substring "w3schools" | 3 |
| [:active](http://www.w3schools.com/cssref/sel_active.asp) | a:active | Selects the active link | 1 |
| [::after](http://www.w3schools.com/cssref/sel_after.asp) | p::after | Insert content after every <p> element | 2 |
| [::before](http://www.w3schools.com/cssref/sel_before.asp) | p::before | Insert content before the content of every <p> element | 2 |
| [:checked](http://www.w3schools.com/cssref/sel_checked.asp) | input:checked | Selects every checked <input> element | 3 |
| [:disabled](http://www.w3schools.com/cssref/sel_disabled.asp) | input:disabled | Selects every disabled <input> element | 3 |
| [:empty](http://www.w3schools.com/cssref/sel_empty.asp) | p:empty | Selects every <p> element that has no children (including text nodes) | 3 |
| [:enabled](http://www.w3schools.com/cssref/sel_enabled.asp) | input:enabled | Selects every enabled <input> element | 3 |
| [:first-child](http://www.w3schools.com/cssref/sel_firstchild.asp) | p:first-child | Selects every <p> element that is the first child of its parent | 2 |
| [::first-letter](http://www.w3schools.com/cssref/sel_firstletter.asp) | p::first-letter | Selects the first letter of every <p> element | 1 |
| [::first-line](http://www.w3schools.com/cssref/sel_firstline.asp) | p::first-line | Selects the first line of every <p> element | 1 |
| [:first-of-type](http://www.w3schools.com/cssref/sel_first-of-type.asp) | p:first-of-type | Selects every <p> element that is the first <p> element of its parent | 3 |
| [:focus](http://www.w3schools.com/cssref/sel_focus.asp) | input:focus | Selects the input element which has focus | 2 |
| [:hover](http://www.w3schools.com/cssref/sel_hover.asp) | a:hover | Selects links on mouse over | 1 |
| [:in-range](http://www.w3schools.com/cssref/sel_in-range.asp) | input:in-range | Selects input elements with a value within a specified range | 3 |
| [:invalid](http://www.w3schools.com/cssref/sel_invalid.asp) | input:invalid | Selects all input elemets with an invalid value | 3 |
| [:lang(language)](http://www.w3schools.com/cssref/sel_lang.asp) | p:lang(it) | Selects every <p> element with a lang attribute equal to "it" (Italian) | 2 |
| [:last-child](http://www.w3schools.com/cssref/sel_last-child.asp) | p:last-child | Selects every <p> element that is the last child of its parent | 3 |
| [:last-of-type](http://www.w3schools.com/cssref/sel_last-of-type.asp) | p:last-of-type | Selects every <p> element that is the last <p> element of its parent | 3 |
| [:link](http://www.w3schools.com/cssref/sel_link.asp) | a:link | Selects all unvisited links | 1 |
| [:not(selector)](http://www.w3schools.com/cssref/sel_not.asp) | :not(p) | Selects every element that is not a <p> element | 3 |
| [:nth-child(n)](http://www.w3schools.com/cssref/sel_nth-child.asp) | p:nth-child(2) | Selects every <p> element that is the second child of its parent | 3 |
| [:nth-last-child(n)](http://www.w3schools.com/cssref/sel_nth-last-child.asp) | p:nth-last-child(2) | Selects every <p> element that is the second child of its parent, counting from the last child | 3 |
| [:nth-last-of-type(n)](http://www.w3schools.com/cssref/sel_nth-last-of-type.asp) | p:nth-last-of-type(2) | Selects every <p> element that is the second <p> element of its parent, counting from the last child | 3 |
| [:nth-of-type(n)](http://www.w3schools.com/cssref/sel_nth-of-type.asp) | p:nth-of-type(2) | Selects every <p> element that is the second <p> element of its parent | 3 |
| [:only-of-type](http://www.w3schools.com/cssref/sel_only-of-type.asp) | p:only-of-type | Selects every <p> element that is the only <p> element of its parent | 3 |
| [:only-child](http://www.w3schools.com/cssref/sel_only-child.asp) | p:only-child | Selects every <p> element that is the only child of its parent | 3 |
| [:optional](http://www.w3schools.com/cssref/sel_optional.asp) | input:optional | Selects input elements with no "required" attribute | 3 |
| [:out-of-range](http://www.w3schools.com/cssref/sel_out-of-range.asp) | input:out-of-range | Selects input elements with a value outside a specified range | 3 |
| [:read-only](http://www.w3schools.com/cssref/sel_read-only.asp) | input:read-only | Selects input elements with the "readonly" attribute specified | 3 |
| [:read-write](http://www.w3schools.com/cssref/sel_read-write.asp) | input:read-write | Selects input elements with the "readonly" attribute NOT specified | 3 |
| [:required](http://www.w3schools.com/cssref/sel_required.asp) | input:required | Selects input elements with the "required" attribute specified | 3 |
| [:root](http://www.w3schools.com/cssref/sel_root.asp) | :root | Selects the document's root element | 3 |
| [::selection](http://www.w3schools.com/cssref/sel_selection.asp) | ::selection | Selects the portion of an element that is selected by a user |  |
| [:target](http://www.w3schools.com/cssref/sel_target.asp) | #news:target | Selects the current active #news element (clicked on a URL containing that anchor name) | 3 |
| [:valid](http://www.w3schools.com/cssref/sel_valid.asp) | input:valid | Selects all input elements with a valid value | 3 |
| [:visited](http://www.w3schools.com/cssref/sel_visited.asp) | a:visited | Selects all visited links | 1 |

|  |  |
| --- | --- |
| **Avoid using these node object properties and methods on HTML document objects:** | |
|  |  |
| **Property / Method** | **Reason for avoiding** |
| document.attributes | Documents don't have attributes |
| document.hasAttributes() | Documents don't have attributes |
| document.nextSibling | Documents don't have siblings |
| document.nodeName | This is always #document |
| document.nodeType | This is always 9 (DOCUMENT\_NODE) |
| document.nodeValue | Documents don't have an node value |
| document.ownerDocument | Documents don't have an owner document |
| document.ownerElement | Documents don't have an owner element |
| document.parentNode | Documents don't have a parent node |
| document.previousSibling | Documents don't have siblings |
| document.textContent | Documents don't have a text content |

# DOM Element

|  |  |
| --- | --- |
| **Property / Method** | **Description** |
| [element.accessKey](http://www.w3schools.com/jsref/prop_html_accesskey.asp) | document.getElementById("myAnchor").accessKey = "w"; |
| [element.addEventListener()](http://www.w3schools.com/jsref/met_element_addeventlistener.asp) | document.getElementById("myBtn").addEventListener("click", function(){     document.getElementById("demo").innerHTML = "Hello World"; }); |
| [element.appendChild()](http://www.w3schools.com/jsref/met_node_appendchild.asp)  .add( selector ), .add( elements ), .add( html ), .add( selection ), .add( selector, context )  .append( content [, content ] )  .append( function( Integer index, String html ) {} ) | var node = document.createElement("LI");  var textnode = document.createTextNode("Water");  node.appendChild(textnode);  document.getElementById("myList").appendChild(node); |
| [element.attributes](http://www.w3schools.com/jsref/prop_node_attributes.asp) | var x = document.getElementById("myBtn").attributes.length; |
| [element.blur()](http://www.w3schools.com/jsref/met_html_blur.asp) | Removes focus from an element |
| [element.childElementCount](http://www.w3schools.com/jsref/prop_element_childelementcount.asp) | Returns the number of child elements an element has |
| [element.childNodes](http://www.w3schools.com/jsref/prop_node_childnodes.asp) | Returns a collection of an element's child nodes (including text and comment nodes) |
| [element.children](http://www.w3schools.com/jsref/prop_element_children.asp) | Returns a collection of an element's child element (excluding text and comment nodes) |
| [element.className](http://www.w3schools.com/jsref/prop_html_classname.asp) | Sets or returns the value of the class attribute of an element |
| [element.click()](http://www.w3schools.com/jsref/met_html_click.asp) | Simulates a mouse-click on an element |
| [element.clientHeight](http://www.w3schools.com/jsref/prop_element_clientheight.asp) | Returns the height of an element, including padding |
| [element.clientLeft](http://www.w3schools.com/jsref/prop_element_clientleft.asp) | Returns the width of the left border of an element |
| [element.clientTop](http://www.w3schools.com/jsref/prop_element_clienttop.asp) | Returns the width of the top border of an element |
| [element.clientWidth](http://www.w3schools.com/jsref/prop_element_clientwidth.asp) | Returns the width of an element, including padding |
| [element.cloneNode()](http://www.w3schools.com/jsref/met_node_clonenode.asp) | var itm = document.getElementById("myList2").lastChild;  var cln = itm.cloneNode(true);  document.getElementById("myList1").appendChild(cln); |
| [element.compareDocumentPosition()](http://www.w3schools.com/jsref/met_node_comparedocumentposition.asp) | var p1 = document.getElementById("p1"); var p2 = document.getElementById("p2"); var x = p1.compareDocumentPosition(p2);  1: No relationship, the two nodes do not belong to the same document.  2: The first node (p1) is positioned after the second node (p2).  4: The first node (p1) is positioned before the second node (p2).  8: The first node (p1) is positioned inside the second node (p2).  16: The second node (p1) is positioned outside the first node (p2).  32: No relationship, or the two nodes are two attributes on the same element. |
| [element.contains()](http://www.w3schools.com/jsref/met_node_contains.asp) | var span = document.getElementById("mySPAN"); var div = document.getElementById("myDIV").contains(span); //true/false |
| [element.contentEditable](http://www.w3schools.com/jsref/prop_html_contenteditable.asp) | document.getElementById("myP").contentEditable = true; |
| [element.dir](http://www.w3schools.com/jsref/prop_html_dir.asp) | document.getElementById("myP").dir = "rtl"; |
| [element.firstChild](http://www.w3schools.com/jsref/prop_node_firstchild.asp) | Returns the first child node of an element |
| [element.firstElementChild](http://www.w3schools.com/jsref/prop_element_firstelementchild.asp) | Returns the first child element of an element |
| [element.focus()](http://www.w3schools.com/jsref/met_html_focus.asp) | Gives focus to an element |
| [element.getAttribute()](http://www.w3schools.com/jsref/met_element_getattribute.asp) | var x = document.getElementsByTagName("H1")[0].getAttribute("class"); |
| [element.getAttributeNode()](http://www.w3schools.com/jsref/met_element_getattributenode.asp) | var elmnt = document.getElementsByTagName("H1")[0]; var attr = elmnt.getAttributeNode("class").value; |
| [element.getElementsByClassName()](http://www.w3schools.com/jsref/met_element_getelementsbyclassname.asp) | Returns a collection of all child elements with the specified class name |
| [element.getElementsByTagName()](http://www.w3schools.com/jsref/met_element_getelementsbytagname.asp) | Returns a collection of all child elements with the specified tag name |
| *element*.getFeature() | Returns an object which implements the APIs of a specified feature |
| [element.hasAttribute()](http://www.w3schools.com/jsref/met_element_hasattribute.asp) | var x = document.getElementById("myBtn").hasAttribute("onclick"); |
| [element.hasAttributes()](http://www.w3schools.com/jsref/met_node_hasattributes.asp) | Returns true if an element has any attributes, otherwise false |
| [element.hasChildNodes()](http://www.w3schools.com/jsref/met_node_haschildnodes.asp) | Returns true if an element has any child nodes, otherwise false |
| [element.id](http://www.w3schools.com/jsref/prop_html_id.asp) | Sets or returns the value of the id attribute of an element |
| [element.innerHTML](http://www.w3schools.com/jsref/prop_html_innerhtml.asp) | Sets or returns the content of an element |
| [element.insertBefore()](http://www.w3schools.com/jsref/met_node_insertbefore.asp) | var newItem = document.createElement("LI");  var textnode = document.createTextNode("Water");  newItem.appendChild(textnode);  var list = document.getElementById("myList");  list.insertBefore(newItem, list.childNodes[0]); |
| [element.isContentEditable](http://www.w3schools.com/jsref/prop_html_iscontenteditable.asp) | Returns true if the content of an element is editable, otherwise false |
| [element.isDefaultNamespace()](http://www.w3schools.com/jsref/met_node_isdefaultnamespace.asp) | var newItem = document.createElement("LI");  var textnode = document.createTextNode("Water");  newItem.appendChild(textnode);  var list = document.getElementById("myList");  list.insertBefore(newItem, list.childNodes[0]); |
| [element.isEqualNode()](http://www.w3schools.com/jsref/met_node_isequalnode.asp) | var item1 = document.getElementById("myList1").firstChild; var item2 = document.getElementById("myList2").firstChild; var x = item1.isEqualNode(item2); |
| [element.isSameNode()](http://www.w3schools.com/jsref/met_node_issamenode.asp) | var item1 = document.getElementById("myList1");  var item2 = document.getElementsByTagName("UL")[0];  var x = item1.isSameNode(item2); |
| [element.isSupported()](http://www.w3schools.com/jsref/met_node_issupported.asp) | var item = document.getElementsByTagName("BUTTON")[0]; item.isSupported("Core", "2.0"); |
| [element.lang](http://www.w3schools.com/jsref/prop_html_lang.asp) | var x = document.getElementById("myP").lang; //en |
| [element.lastChild](http://www.w3schools.com/jsref/prop_node_lastchild.asp) | Returns the last child node of an element |
| [element.lastElementChild](http://www.w3schools.com/jsref/prop_element_lastelementchild.asp) | Returns the last child element of an element |
| [element.namespaceURI](http://www.w3schools.com/jsref/prop_node_namespaceuri.asp) | var x = document.documentElement.namespaceURI; //http://www.w3.org/1999/xhtml |
| [element.nextSibling](http://www.w3schools.com/jsref/prop_node_nextsibling.asp) | Returns the next node at the same node tree level |
| [element.nextElementSibling](http://www.w3schools.com/jsref/prop_element_nextelementsibling.asp) | Returns the next element at the same node tree level |
| [element.nodeName](http://www.w3schools.com/jsref/prop_node_nodename.asp) | var x = document.getElementById("myP").nodeName; //P |
| [element.nodeType](http://www.w3schools.com/jsref/prop_node_nodetype.asp) | var x = document.getElementById("myP").nodeType;  If the node is an element node, the nodeType property will return 1.  If the node is an attribute node, the nodeType property will return 2.  If the node is a text node, the nodeType property will return 3.  If the node is a comment node, the nodeType property will return 8. |
| [element.nodeValue](http://www.w3schools.com/jsref/prop_node_nodevalue.asp) | var x = document.getElementsByTagName("BUTTON")[0].childNodes[0].nodeValue; /\* Try it yourself \*/ |
| [element.normalize()](http://www.w3schools.com/jsref/met_node_normalize.asp) | Joins adjacent text nodes and removes empty text nodes in an element |
| [element.offsetHeight](http://www.w3schools.com/jsref/prop_element_offsetheight.asp) | Returns the height of an element, including padding, **border and scrollbar** |
| [element.offsetWidth](http://www.w3schools.com/jsref/prop_element_offsetwidth.asp) | Returns the width of an element, including padding, **border and scrollbar** |
| *element*.offsetLeft | Returns the horizontal offset position of an element |
| *element*.offsetParent | Returns the offset container of an element |
| *element*.offsetTop | Returns the vertical offset position of an element |
| [element.ownerDocument](http://www.w3schools.com/jsref/prop_node_ownerdocument.asp) | Returns the root element (document object) for an element |
| [element.parentNode](http://www.w3schools.com/jsref/prop_node_parentnode.asp) | Returns the parent node of an element |
| [element.parentElement](http://www.w3schools.com/jsref/prop_node_parentelement.asp) | Returns the parent element node of an element |
| [element.previousSibling](http://www.w3schools.com/jsref/prop_node_previoussibling.asp) | Returns the previous node at the same node tree level |
| [element.previousElementSibling](http://www.w3schools.com/jsref/prop_element_previouselementsibling.asp) | Returns the previous element at the same node tree level |
| [element.querySelector()](http://www.w3schools.com/jsref/met_element_queryselector.asp) | Returns the first child element that matches a specified CSS selector(s) of an element |
| [element.querySelectorAll()](http://www.w3schools.com/jsref/met_element_queryselectorall.asp) | Returns all child elements that matches a specified CSS selector(s) of an element |
| [element.removeAttribute()](http://www.w3schools.com/jsref/met_element_removeattribute.asp) | Removes a specified attribute from an element |
| [element.removeAttributeNode()](http://www.w3schools.com/jsref/met_element_removeattributenode.asp) | Removes a specified attribute node, and returns the removed node |
| [element.removeChild()](http://www.w3schools.com/jsref/met_node_removechild.asp) | Removes a child node from an element |
| [element.replaceChild()](http://www.w3schools.com/jsref/met_node_replacechild.asp) | Replaces a child node in an element |
| [element.removeEventListener()](http://www.w3schools.com/jsref/met_element_removeeventlistener.asp) | document.getElementById("myDIV").removeEventListener("mousemove", myFunction); |
| [element.scrollHeight](http://www.w3schools.com/jsref/prop_element_scrollheight.asp) | Returns the entire height of an element, including padding |
| [element.scrollLeft](http://www.w3schools.com/jsref/prop_element_scrollleft.asp) | Sets or returns the number of pixels an element's content is scrolled horizontally |
| [element.scrollTop](http://www.w3schools.com/jsref/prop_element_scrolltop.asp) | Sets or returns the number of pixels an element's content is scrolled vertically |
| [element.scrollWidth](http://www.w3schools.com/jsref/prop_element_scrollwidth.asp) | Returns the entire width of an element, including padding |
| [element.setAttribute()](http://www.w3schools.com/jsref/met_element_setattribute.asp) | Sets or changes the specified attribute, to the specified value |
| [element.setAttributeNode()](http://www.w3schools.com/jsref/met_element_setattributenode.asp) | Sets or changes the specified attribute node |
| [element.style](http://www.w3schools.com/jsref/prop_html_style.asp) | Sets or returns the value of the style attribute of an element |
| [element.tabIndex](http://www.w3schools.com/jsref/prop_html_tabindex.asp) | Sets or returns the value of the tabindex attribute of an element |
| [element.tagName](http://www.w3schools.com/jsref/prop_element_tagname.asp) | Returns the tag name of an element |
| [element.textContent](http://www.w3schools.com/jsref/prop_node_textcontent.asp) | Sets or returns the textual content of a node and its descendants |
| [element.title](http://www.w3schools.com/jsref/prop_html_title.asp) | Sets or returns the value of the title attribute of an element |
| *element*.toString() | Converts an element to a string |
|  |  |
| [nodelist.item()](http://www.w3schools.com/jsref/met_nodelist_item.asp) | Returns the node at the specified index in a NodeList |
| [nodelist.length](http://www.w3schools.com/jsref/prop_nodelist_length.asp) | Returns the number of nodes in a NodeList |

# DOM Attribute

|  |  |
| --- | --- |
| **Property / Method** | **Description** |
| [attr.isId](http://www.w3schools.com/jsref/prop_attr_isid.asp) | Returns true if the attribute is of type Id, otherwise it returns false |
| [attr.name](http://www.w3schools.com/jsref/prop_attr_name.asp) | Returns the name of an attribute |
| [attr.value](http://www.w3schools.com/jsref/prop_attr_value.asp) | Sets or returns the value of the attribute |
| [attr.specified](http://www.w3schools.com/jsref/prop_attr_specified.asp) | var x = document.getElementById("demo").attributes[0].specified; //true/false |
|  |  |
| [nodemap.getNamedItem()](http://www.w3schools.com/jsref/met_namednodemap_getnameditem.asp) | var btn = document.getElementsByTagName("BUTTON")[0]; btn.attributes.getNamedItem("onclick").textContent; //myfunction() |
| [nodemap.item()](http://www.w3schools.com/jsref/met_namednodemap_item.asp) | var x = document.getElementsByTagName("BUTTON")[0].attributes.item(0).nodeName; //onclick |
| [nodemap.length](http://www.w3schools.com/jsref/prop_namednodemap_length.asp) | Returns the number of attribute nodes in a NamedNodeMap |
| [nodemap.removeNamedItem()](http://www.w3schools.com/jsref/met_namednodemap_removenameditem.asp) | var btn = document.getElementsByTagName("INPUT")[0];  btn.attributes.removeNamedItem("type"); |
| [nodemap.setNamedItem()](http://www.w3schools.com/jsref/met_namednodemap_setnameditem.asp) | var btn = document.getElementsByTagName("H1")[0]; var typ = document.createAttribute("class"); typ.value = "democlass"; btn.attributes.setNamedItem(typ); |

|  |  |
| --- | --- |
| **For future code quality, you should avoid using node object properties and methods on attribute objects:** | |
|  |  |
| **Property / Method** | **Reason for avoiding** |
| *attr*.appendChild() | Attributes don't have child nodes |
| *attr*.attributes | Attributes don't have attributes |
| *attr*.baseURI | use document.baseURI instead |
| *attr*.childNodes | Attributes don't have child nodes |
| *attr*.cloneNode() | Get or set the *attr*.value instead |
| *attr*.firstChild | Attributes don't have child nodes |
| *attr*.hasAttributes() | Attributes don't have attributes |
| *attr*.hasChildNodes | Attributes don't have child nodes |
| *attr*.insertBefore() | Attributes don't have child nodes |
| *attr*.isEqualNode() | Makes no sense |
| *attr*.isSameNode() | Makes no sense |
| *attr*.isSupported() | Is always true |
| *attr*.lastChild | Attributes don't have child nodes |
| *attr*.nextSibling | Attributes don't have siblings |
| *attr*.nodeName | Use *attr*.name instead |
| *attr*.nodeType | This is always 2 (ATTRIBUTE\_NODE) |
| *attr*.nodeValue | Use *attr*.value instead |
| *attr*.normalize() | Attributes cannot be normalized |
| *attr*.ownerDocument | This is always your HTML document |
| *attr*.ownerElement | This is the HTML element you used to access the attribute |
| *attr*.parentNode | This is the HTML element you used to access the attribute |
| *attr*.previousSibling | Attributes don't have siblings |
| *attr*.removeChild | Attributes don't have child nodes |
| *attr*.replaceChild | Attributes don't have child nodes |
| *attr*.textContent | Use *attr*.value instead |

# DOM Events

|  |  |  |
| --- | --- | --- |
| **Mouse Events** |  |  |
|  |  |  |
| **Event** | **Description** | **DOM** |
| [onclick](http://www.w3schools.com/jsref/event_onclick.asp) | The event occurs when the user clicks on an element | 2 |
| [oncontextmenu](http://www.w3schools.com/jsref/event_oncontextmenu.asp) | The event occurs when the user right-clicks on an element to open a context menu | 3 |
| [ondblclick](http://www.w3schools.com/jsref/event_ondblclick.asp) | The event occurs when the user double-clicks on an element | 2 |
| [onmousedown](http://www.w3schools.com/jsref/event_onmousedown.asp) | The event occurs when the user presses a mouse button over an element | 2 |
| [onmouseenter](http://www.w3schools.com/jsref/event_onmouseenter.asp) | The event occurs when the pointer is moved onto an element | 2 |
| [onmouseleave](http://www.w3schools.com/jsref/event_onmouseleave.asp) | The event occurs when the pointer is moved out of an element | 2 |
| [onmousemove](http://www.w3schools.com/jsref/event_onmousemove.asp) | The event occurs when the pointer is moving while it is over an element | 2 |
| [onmouseover](http://www.w3schools.com/jsref/event_onmouseover.asp) | The event occurs when the pointer is moved onto an element, or onto one of its children | 2 |
| [onmouseout](http://www.w3schools.com/jsref/event_onmouseout.asp) | The event occurs when a user moves the mouse pointer out of an element, or out of one of its children | 2 |
| [onmouseup](http://www.w3schools.com/jsref/event_onmouseup.asp) | The event occurs when a user releases a mouse button over an element | 2 |
|  |  |  |
| **Keyboard Events** |  |  |
|  |  |  |
| **Event** | **Description** | **DOM** |
| [onkeydown](http://www.w3schools.com/jsref/event_onkeydown.asp) | The event occurs when the user is pressing a key | 2 |
| [onkeypress](http://www.w3schools.com/jsref/event_onkeypress.asp) | The event occurs when the user presses a key | 2 |
| [onkeyup](http://www.w3schools.com/jsref/event_onkeyup.asp) | The event occurs when the user releases a key | 2 |
|  |  |  |
| **Frame/Object Events** |  |  |
|  |  |  |
| **Event** | **Description** | **DOM** |
| [onabort](http://www.w3schools.com/jsref/event_onabort.asp) | The event occurs when the loading of a resource has been aborted | 2 |
| [onbeforeunload](http://www.w3schools.com/jsref/event_onbeforeunload.asp) | The event occurs before the document is about to be unloaded | 2 |
| [onerror](http://www.w3schools.com/jsref/event_onerror.asp) | The event occurs when an error occurs while loading an external file | 2 |
| [onhashchange](http://www.w3schools.com/jsref/event_onhashchange.asp) | The event occurs when there has been changes to the anchor part of a URL | 3 |
| [onload](http://www.w3schools.com/jsref/event_onload.asp) | The event occurs when an object has loaded | 2 |
| [onpageshow](http://www.w3schools.com/jsref/event_onpageshow.asp) | The event occurs when the user navigates to a webpage | 3 |
| onpagehide | The event occurs when the user navigates away from a webpage | 3 |
| [onresize](http://www.w3schools.com/jsref/event_onresize.asp) | The event occurs when the document view is resized | 2 |
| [onscroll](http://www.w3schools.com/jsref/event_onscroll.asp) | The event occurs when an element's scrollbar is being scrolled | 2 |
| [onunload](http://www.w3schools.com/jsref/event_onunload.asp) | The event occurs once a page has unloaded (for <body>) | 2 |
|  |  |  |
| **Form Events** |  |  |
|  |  |  |
| **Event** | **Description** | **DOM** |
| [onblur](http://www.w3schools.com/jsref/event_onblur.asp) | The event occurs when an element loses focus | 2 |
| [onchange](http://www.w3schools.com/jsref/event_onchange.asp) | The event occurs when the content of a form element, the selection, or the checked state have changed (for <input>, <keygen>, <select>, and <textarea>) | 2 |
| [onfocus](http://www.w3schools.com/jsref/event_onfocus.asp) | The event occurs when an element gets focus | 2 |
| [onfocusin](http://www.w3schools.com/jsref/event_onfocusin.asp) | The event occurs when an element is about to get focus | 2 |
| [onfocusout](http://www.w3schools.com/jsref/event_onfocusout.asp) | The event occurs when an element is about to lose focus | 2 |
| [oninput](http://www.w3schools.com/jsref/event_oninput.asp) | The event occurs when an element gets user input | 3 |
| oninvalid | The event occurs when an element is invalid | 3 |
| [onreset](http://www.w3schools.com/jsref/event_onreset.asp) | The event occurs when a form is reset | 2 |
| [onsearch](http://www.w3schools.com/jsref/event_onsearch.asp) | The event occurs when the user writes something in a search field (for <input="search">) | 3 |
| [onselect](http://www.w3schools.com/jsref/event_onselect.asp) | The event occurs after the user selects some text (for <input> and <textarea>) | 2 |
| [onsubmit](http://www.w3schools.com/jsref/event_onsubmit.asp) | The event occurs when a form is submitted | 2 |
|  |  |  |
| **Drag Events** |  |  |
|  |  |  |
| **Event** | **Description** | **DOM** |
| [ondrag](http://www.w3schools.com/jsref/event_ondrag.asp) | The event occurs when an element is being dragged | 3 |
| [ondragend](http://www.w3schools.com/jsref/event_ondragend.asp) | The event occurs when the user has finished dragging an element | 3 |
| [ondragenter](http://www.w3schools.com/jsref/event_ondragenter.asp) | The event occurs when the dragged element enters the drop target | 3 |
| [ondragleave](http://www.w3schools.com/jsref/event_ondragleave.asp) | The event occurs when the dragged element leaves the drop target | 3 |
| [ondragover](http://www.w3schools.com/jsref/event_ondragover.asp) | The event occurs when the dragged element is over the drop target | 3 |
| [ondragstart](http://www.w3schools.com/jsref/event_ondragstart.asp) | The event occurs when the user starts to drag an element | 3 |
| [ondrop](http://www.w3schools.com/jsref/event_ondrop.asp) | The event occurs when the dragged element is dropped on the drop target | 3 |
|  |  |  |
| **Clipboard Events** |  |  |
|  |  |  |
| **Event** | **Description** | **DOM** |
| [oncopy](http://www.w3schools.com/jsref/event_oncopy.asp) | The event occurs when the user copies the content of an element |  |
| [oncut](http://www.w3schools.com/jsref/event_oncut.asp) | The event occurs when the user cuts the content of an element |  |
| [onpaste](http://www.w3schools.com/jsref/event_onpaste.asp) | The event occurs when the user pastes some content in an element |  |
|  |  |  |
| **Print Events** |  |  |
|  |  |  |
| **Event** | **Description** | **DOM** |
| [onafterprint](http://www.w3schools.com/jsref/event_onafterprint.asp) | The event occurs when a page has started printing, or if the print dialogue box has been closed | 3 |
| [onbeforeprint](http://www.w3schools.com/jsref/event_onbeforeprint.asp) | The event occurs when a page is about to be printed | 3 |
|  |  |  |
| **Media Events** |  |  |
|  |  |  |
| **Event** | **Description** | **DOM** |
| [onabort](http://www.w3schools.com/jsref/event_onabort_media.asp) | The event occurs when the loading of a media is aborted | 3 |
| [oncanplay](http://www.w3schools.com/jsref/event_oncanplay.asp) | The event occurs when the browser can start playing the media (when it has buffered enough to begin) | 3 |
| [oncanplaythrough](http://www.w3schools.com/jsref/event_oncanplaythrough.asp) | The event occurs when the browser can play through the media without stopping for buffering | 3 |
| [ondurationchange](http://www.w3schools.com/jsref/event_ondurationchange.asp) | The event occurs when the duration of the media is changed | 3 |
| onemptied | The event occurs when something bad happens and the media file is suddenly unavailable (like unexpectedly disconnects) | 3 |
| [onended](http://www.w3schools.com/jsref/event_onended.asp) | The event occurs when the media has reach the end (useful for messages like "thanks for listening") | 3 |
| [onerror](http://www.w3schools.com/jsref/event_onerror_media.asp) | The event occurs when an error occurred during the loading of a media file | 3 |
| [onloadeddata](http://www.w3schools.com/jsref/event_onloadeddata.asp) | The event occurs when media data is loaded | 3 |
| [onloadedmetadata](http://www.w3schools.com/jsref/event_onloadedmetadata.asp) | The event occurs when meta data (like dimensions and duration) are loaded | 3 |
| [onloadstart](http://www.w3schools.com/jsref/event_onloadstart.asp) | The event occurs when the browser starts looking for the specified media | 3 |
| [onpause](http://www.w3schools.com/jsref/event_onpause.asp) | The event occurs when the media is paused either by the user or programmatically | 3 |
| [onplay](http://www.w3schools.com/jsref/event_onplay.asp) | The event occurs when the media has been started or is no longer paused | 3 |
| [onplaying](http://www.w3schools.com/jsref/event_onplaying.asp) | The event occurs when the media is playing after having been paused or stopped for buffering | 3 |
| [onprogress](http://www.w3schools.com/jsref/event_onprogress.asp) | The event occurs when the browser is in the process of getting the media data (downloading the media) | 3 |
| [onratechange](http://www.w3schools.com/jsref/event_onratechange.asp) | The event occurs when the playing speed of the media is changed | 3 |
| [onseeked](http://www.w3schools.com/jsref/event_onseeked.asp) | The event occurs when the user is finished moving/skipping to a new position in the media | 3 |
| [onseeking](http://www.w3schools.com/jsref/event_onseeking.asp) | The event occurs when the user starts moving/skipping to a new position in the media | 3 |
| [onstalled](http://www.w3schools.com/jsref/event_onstalled.asp) | The event occurs when the browser is trying to get media data, but data is not available | 3 |
| [onsuspend](http://www.w3schools.com/jsref/event_onsuspend.asp) | The event occurs when the browser is intentionally not getting media data | 3 |
| [ontimeupdate](http://www.w3schools.com/jsref/event_ontimeupdate.asp) | The event occurs when the playing position has changed (like when the user fast forwards to a different point in the media) | 3 |
| [onvolumechange](http://www.w3schools.com/jsref/event_onvolumechange.asp) | The event occurs when the volume of the media has changed (includes setting the volume to "mute") | 3 |
| [onwaiting](http://www.w3schools.com/jsref/event_onwaiting.asp) | The event occurs when the media has paused but is expected to resume (like when the media pauses to buffer more data) | 3 |
|  |  |  |
| **Animation Events** |  |  |
|  |  |  |
| **Event** | **Description** | **DOM** |
| [animationend](http://www.w3schools.com/jsref/event_animationend.asp) | The event occurs when a CSS animation has completed | 3 |
| [animationiteration](http://www.w3schools.com/jsref/event_animationiteration.asp) | The event occurs when a CSS animation is repeated | 3 |
| [animationstart](http://www.w3schools.com/jsref/event_animationstart.asp) | The event occurs when a CSS animation has started | 3 |
|  |  |  |
| **Transition Events** |  |  |
|  |  |  |
| **Event** | **Description** | **DOM** |
| [transitionend](http://www.w3schools.com/jsref/event_transitionend.asp) | The event occurs when a CSS transition has completed | 3 |
|  |  |  |
| **Misc Events** |  |  |
|  |  |  |
| **Event** | **Description** | **DOM** |
| onmessage | The event occurs when a message is received through or from an object (WebSocket, Web Worker, Event Source or a child frame or a parent window) | 3 |
| onmousewheel | [Deprecated. Use the onwheel event instead](http://www.w3schools.com/jsref/event_onwheel.asp) |  |
| [ononline](http://www.w3schools.com/jsref/event_ononline.asp) | The event occurs when the browser starts to work online | 3 |
| [onoffline](http://www.w3schools.com/jsref/event_onoffline.asp) | The event occurs when the browser starts to work offline | 3 |
| onpopstate | The event occurs when the window's history changes | 3 |
| [onshow](http://www.w3schools.com/jsref/event_onshow.asp) | The event occurs when a <menu> element is shown as a context menu | 3 |
| onstorage | The event occurs when a Web Storage area is updated | 3 |
| [ontoggle](http://www.w3schools.com/jsref/event_ontoggle.asp) | The event occurs when the user opens or closes the <details> element | 3 |
| [onwheel](http://www.w3schools.com/jsref/event_onwheel.asp) | The event occurs when the mouse wheel rolls up or down over an element | 3 |
|  |  |  |
| **Event Object** |  |  |
|  |  |  |
| **Constants** |  |  |
|  |  |  |
| **Constant** | **Description** | **DOM** |
| CAPTURING\_PHASE | The current event phase is the capture phase (1) | 1 |
| AT\_TARGET | The current event is in the target phase, i.e. it is being evaluated at the event target (2) | 2 |
| BUBBLING\_PHASE | The current event phase is the bubbling phase (3) | 3 |
|  |  |  |
| **Properties** |  |  |
|  |  |  |
| **Property** | **Description** | **DOM** |
| [bubbles](http://www.w3schools.com/jsref/event_bubbles.asp) | Returns whether or not a specific event is a bubbling event | 2 |
| [cancelable](http://www.w3schools.com/jsref/event_cancelable.asp) | Returns whether or not an event can have its default action prevented | 2 |
| [currentTarget](http://www.w3schools.com/jsref/event_currenttarget.asp) | Returns the element whose event listeners triggered the event | 2 |
| [defaultPrevented](http://www.w3schools.com/jsref/event_defaultprevented.asp) | Returns whether or not the preventDefault() method was called for the event | 3 |
| [eventPhase](http://www.w3schools.com/jsref/event_eventphase.asp) | Returns which phase of the event flow is currently being evaluated | 2 |
| [isTrusted](http://www.w3schools.com/jsref/event_istrusted.asp) | Returns whether or not an event is trusted | 3 |
| [target](http://www.w3schools.com/jsref/event_target.asp) | Returns the element that triggered the event | 2 |
| [timeStamp](http://www.w3schools.com/jsref/event_timestamp.asp) | Returns the time (in milliseconds relative to the epoch) at which the event was created | 2 |
| [type](http://www.w3schools.com/jsref/event_type.asp) | Returns the name of the event | 2 |
| [view](http://www.w3schools.com/jsref/event_view.asp) | Returns a reference to the Window object where the event occured | 2 |
|  |  |  |
| **Methods** |  |  |
|  |  |  |
| **Method** | **Description** | **DOM** |
| initEvent() | Specifies the event type, whether or not the event can bubble, whether or not the event's default action can be prevented | 2 |
| [preventDefault()](http://www.w3schools.com/jsref/event_preventdefault.asp) | Cancels the event if it is cancelable, meaning that the default action that belongs to the event will not occur | 2 |
| stopImmediatePropagation() | Prevents other listeners of the same event from being called | 3 |
| stopPropagation() | Prevents further propagation of an event during event flow | 2 |
|  |  |  |
| **MouseEvent Object** |  |  |
|  |  |  |
| **Property** | **Description** | **DOM** |
| [altKey](http://www.w3schools.com/jsref/event_altkey.asp) | Returns whether the "ALT" key was pressed when the mouse event was triggered | 2 |
| [button](http://www.w3schools.com/jsref/event_button.asp) | Returns which mouse button was pressed when the mouse event was triggered | 2 |
| [buttons](http://www.w3schools.com/jsref/event_buttons.asp) | Returns which mouse buttons were pressed when the mouse event was triggered | 3 |
| [clientX](http://www.w3schools.com/jsref/event_clientx.asp) | Returns the horizontal coordinate of the mouse pointer, relative to the current window, when the mouse event was triggered | 2 |
| [clientY](http://www.w3schools.com/jsref/event_clienty.asp) | Returns the vertical coordinate of the mouse pointer, relative to the current window, when the mouse event was triggered | 2 |
| [ctrlKey](http://www.w3schools.com/jsref/event_ctrlkey.asp) | Returns whether the "CTRL" key was pressed when the mouse event was triggered | 2 |
| [detail](http://www.w3schools.com/jsref/event_detail.asp) | Returns a number that indicates how many times the mouse was clicked | 2 |
| [metaKey](http://www.w3schools.com/jsref/event_metakey.asp) | Returns whether the "META" key was pressed when an event was triggered | 2 |
| [relatedTarget](http://www.w3schools.com/jsref/event_relatedtarget.asp) | Returns the element related to the element that triggered the mouse event | 2 |
| [screenX](http://www.w3schools.com/jsref/event_screenx.asp) | Returns the horizontal coordinate of the mouse pointer, relative to the screen, when an event was triggered | 2 |
| [screenY](http://www.w3schools.com/jsref/event_screeny.asp) | Returns the vertical coordinate of the mouse pointer, relative to the screen, when an event was triggered | 2 |
| [shiftKey](http://www.w3schools.com/jsref/event_shiftkey.asp) | Returns whether the "SHIFT" key was pressed when an event was triggered | 2 |
| [which](http://www.w3schools.com/jsref/event_which.asp) | Returns which mouse button was pressed when the mouse event was triggered | 2 |
|  |  |  |
| **KeyboardEvent Object** |  |  |
|  |  |  |
| **Property** | **Description** | **DOM** |
| [altKey](http://www.w3schools.com/jsref/event_key_altkey.asp) | Returns whether the "ALT" key was pressed when the key event was triggered | 2 |
| [ctrlKey](http://www.w3schools.com/jsref/event_key_ctrlkey.asp) | Returns whether the "CTRL" key was pressed when the key event was triggered | 2 |
| [charCode](http://www.w3schools.com/jsref/event_key_charcode.asp) | Returns the Unicode character code of the key that triggered the onkeypress event | 2 |
| [key](http://www.w3schools.com/jsref/event_key_key.asp) | Returns the key value of the key represented by the event | 3 |
| [keyCode](http://www.w3schools.com/jsref/event_key_keycode.asp) | Returns the Unicode character code of the key that triggered the onkeypress event, or the Unicode key code of the key that triggered the onkeydown or onkeyup event | 2 |
| [location](http://www.w3schools.com/jsref/event_key_location.asp) | Returns the location of a key on the keyboard or device | 3 |
| [metaKey](http://www.w3schools.com/jsref/event_key_metakey.asp) | Returns whether the "meta" key was pressed when the key event was triggered | 2 |
| [shiftKey](http://www.w3schools.com/jsref/event_key_shiftkey.asp) | Returns whether the "SHIFT" key was pressed when the key event was triggered | 2 |
| [which](http://www.w3schools.com/jsref/event_key_which.asp) | Returns the Unicode character code of the key that triggered the onkeypress event, or the Unicode key code of the key that triggered the onkeydown or onkeyup event | 2 |
|  |  |  |
| **HashChangeEvent Object** |  |  |
|  |  |  |
| **Property** | **Description** | **DOM** |
| [newURL](http://www.w3schools.com/jsref/event_hashchange_newurl.asp) | Returns the URL of the document, after the hash has been changed |  |
| [oldURL](http://www.w3schools.com/jsref/event_hashchange_oldurl.asp) | Returns the URL of the document, before the hash was changed |  |
|  |  |  |
| **PageTransitionEvent Object** |  |  |
|  |  |  |
| **Property** | **Description** | **DOM** |
| [persisted](http://www.w3schools.com/jsref/event_pagetransition_persisted.asp) | Returns whether the webpage was cached by the browser |  |
|  |  |  |
| **FocusEvent Object** |  |  |
|  |  |  |
| **Property** | **Description** | **DOM** |
| [relatedTarget](http://www.w3schools.com/jsref/event_focus_relatedtarget.asp) | Returns the element related to the element that triggered the event | 3 |
|  |  |  |
| **AnimationEvent Object** |  |  |
|  |  |  |
| **Property** | **Description** | **DOM** |
| [animationName](http://www.w3schools.com/jsref/event_animation_animationName.asp) | Returns the name of the animation |  |
| [elapsedTime](http://www.w3schools.com/jsref/event_animation_elapsedtime.asp) | Returns the number of seconds an animation has been running |  |
|  |  |  |
| **TransitionEvent Object** |  |  |
|  |  |  |
| **Property** | **Description** | **DOM** |
| [propertyName](http://www.w3schools.com/jsref/event_transition_propertyName.asp) | Returns the name of the CSS property associated with the transition |  |
| [elapsedTime](http://www.w3schools.com/jsref/event_transition_elapsedtime.asp) | Returns the number of seconds a transition has been running |  |
|  |  |  |
| **WheelEvent Object** |  |  |
|  |  |  |
| **Property** | **Description** | **DOM** |
| deltaX | Returns the horizontal scroll amount of a mouse wheel (x-axis) | 3 |
| deltaY | Returns the vertical scroll amount of a mouse wheel (y-axis) | 3 |
| deltaZ | Returns the scroll amount of a mouse wheel for the z-axis | 3 |
| deltaMode | Returns a number that represents the unit of measurements for delta values (pixels, lines or pages) | 3 |

Stop Event Propagation**:**

e.preventDefault(); event.stopPropagation();

For IE: window.event.cancelBubble = true

# jQuery Selectors

$('#id'), $('.class'), $('name'), $('\*'), $('class.class'), $('selector1, selector2, ... selectorN'),

$('ancestor descendant'), $('parent > child'), $('prev + next'), $('prev ~ siblings'),

$('tr:first'), $('selector:last'), $(':not(selector)'), ( "input:not(:checked) + span" ), $('sel:even'),

:odd, :eq(index), :gt(index), :lt(index), :header (h1, h2, h3 etc), :animated, :contains(text)

:empty (no children), :parent (has children) :has(selector) (has one child that match selector)

[@attribute], [@attribute=value], [@attribute!=value], [@attribute^=value] (starts with), [@attribute$=value] (ends with), [@attribute\*=value] (contains), [attributeFilter1][attributeFilter2]...

:nth-child(index/even/odd/equation), :first-child, :last-child, :only-child

:hidden, :visible, :input, :text, :password, :radio, :checkbox, :submit, :image, :reset, :button, :file

:enabled, :disabled, :checked, :selected

# jQuery Misc

var $j = jQuery.noConflict();

// $j is now an alias to the jQuery function; creating the new alias is optional.

$j(document).ready(function() {

$j( "div" ).hide();

});

// The $ variable now has the prototype meaning, which is a shortcut for

// document.getElementById(). mainDiv below is a DOM element, not a jQuery object.

var mainDiv = $( "main" );

}

# CSS POSITIONING

static (default), absolute (first absolute parent), relative, fixed(browser window, inherit

# HOW TO DISPLAY DIVs SIDE BY SIDE

|  |
| --- |
| <html>  <head></head>  <body>  <div style="float:left;background-color:red;width:50px;height:20px;"></div>  <div style="float:left;background-color:blue;width:50px;height:20px;"></div>  </body>  </html> |

# HOW TO CREATE A TABLE USING UL AND LI TAGS

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset=utf-8 />  <title>JS Bin</title>  <style>  .d-table {  min-width: 300px;  min-height: 300px;  background: lightgrey;  display: block;  width: 1px solid red;  }  .d-table ul {  list-style-type: none;  }  .d-column li {  padding: 0px;  display: inline-block;  border: 1px solid blue;  background: grey;  width: 50px;  height: 20px;  }  .d-row li {  padding: 0px;  display: inline-block;  background: lightgrey;  border: 1px solid red;  width: 50px;  height: 20px;  }  </style>  </head>  <body>  <div class="d-table">  <ul class="d-column">  <li> </li>  <li> </li>  <li> </li>  <li> </li>  </ul>  <ul class="d-row">  <li> </li>  <li> </li>  <li> </li>  <li> </li>  </ul>  <ul class="d-row">  <li> </li>  <li> </li>  <li> </li>  <li> </li>  </ul>  </div>  </body>  </html> |

# SAMPLE JQUERY PLUGIN

|  |
| --- |
| //jquery required. autocomplete  var \_debugac = true;  (function($){  $.fn.autocomp = function(source, options) {  var prefctrl = 'jautocomp'; var prefitem = 'jacrec\_'; var hlColor = '#99CCFF'; var unhlColor = '#FFFFF0';  var itemcssopt = {'clear': 'right', 'cursor': 'pointer', 'white-space': 'nowrap', 'overflow': 'hidden', 'padding': '0'};  var config = $.extend({}, $.fn.autocomp.defaults, {source: source}, options);  var maxheight = config.maxheight ? config.maxheight : 200;  var elemid = $(this).attr('id'); var elem = $('#'+elemid);  var containerid = elemid+prefctrl; var container = $('#'+containerid);  if (container.length<=0) {  container = $('<div />').attr('id', containerid)  .css({'position': 'absolute', 'display': 'none', 'background-color': '#ffffff', 'border': '1px solid #cccccc',  'overflow-y': 'scroll', 'scrollbar-track-color': '#ffffff', 'scrollbar-face-color': '#ffffff',  'scrollbar-arrow-color': '#000000', 'scrollbar-3dlight-color': '#ffffff', 'scrollbar-highlight-color': '#cccccc',  'scrollbar-shadow-color': '#000000', 'scrollbar-darkshadow-color': '#cccccc', 'padding': '0'});  $('body').append($(container));  linehgt = parseInt(container.css('font-size'))+2;  }  var linehgt = parseInt(elem.css('font-size'))+2;  var cursor = -1; var ctrldata = null; var arrec = new Array(); var contentHeight = 0; var padding = 8; var acTimer = null;  return this.each(function() {  $(this).bind('keyup', function(e){  e = e || window.event;  switch(e.keyCode) {  case 27: //ESC  if (container.length>0) {  e.preventDefault(); e.stopPropagation(); container.css('display', 'none');  }  break;  case 38: //up  if (container.length>0 && container.css('display')=='inline' && ctrldata && ctrldata.length>0) {  cursor = cursor<1 ? 0 : cursor-1;  $(this).val(ctrldata[cursor]['display']);  for (var i=0; i<arrec.length; i++) {  if (arrec[i].length>0) arrec[i].css('background-color', i==cursor ? hlColor : unhlColor);  }  var paddingTop = 1;  try {  paddingTop = parseInt(container.css('padding-top'));  } catch(e) {}  var curbottom = container.scrollTop() + paddingTop + container.height();  var curitemtop = contentHeight \* ((cursor)/arrec.length);  if (curitemtop < container.scrollTop()) {  container.scrollTop(container.scrollTop() - linehgt);  } else if (curitemtop > curbottom) {  container.scrollTop(container.scrollTop() + linehgt);  }  }  break;    case 40: //down  if (container.length>0 && container.css('display')=='inline' && ctrldata && ctrldata.length>0) {  cursor = cursor>=ctrldata.length-1 ? ctrldata.length-1 : cursor+1;  $(this).val(ctrldata[cursor]['display']);  for (var i=0; i<arrec.length; i++) {  if (arrec[i].length>0) arrec[i].css('background-color', i==cursor ? hlColor : unhlColor);  }  var paddingTop = 1;  try {  paddingTop = parseInt(container.css('padding-top'));  } catch(e) {}  var curbottom = container.scrollTop() + paddingTop + container.height();  var curitemtop = contentHeight \* ((cursor+1)/arrec.length);  if (curitemtop < container.scrollTop()) {  container.scrollTop(container.scrollTop() - linehgt);  } else if (curitemtop > curbottom) {  container.scrollTop(container.scrollTop() + linehgt);  }  }  break;  case 13: //enter  if (container.length>0 && container.css('display')=='inline' && ctrldata && ctrldata.length>0 &&  arrec && arrec.length > cursor && cursor > -1 && arrec[cursor]) {  arrec[cursor].click();  //alert('cursor='+cursor);  }  break;  default:  if ($(this).val().length < config.chars) return;  var processing = e.keyCode >= 48 && e.keyCode <= 57 || e.keyCode >= 65 && e.keyCode <= 90 ||  e.keyCode >= 97 && e.keyCode <= 122 || e.keyCode == 32;  if (!processing) return; if (acTimer) clearTimeout(acTimer);  acTimer = setTimeout(function() {  var qry = config.source + escape(elem.val());  elem.attr('disabled', true);  if (config.indicator && config.indicator.length>0) config.indicator.css('display','inline');  $.ajax({url: qry,  dataType: 'json',  success: function(data) {  elem.attr('disabled', false).focus();  if (config.indicator && config.indicator.length>0) config.indicator.css('display','none');  if (!acTimer) return;  cursor = -1; ctrldata = data.length>0 ? data : [{"value":"","display":""}]; container.html(''); arrec = new Array();  if (data.length<1) {  var tmprec = $('<div />').html('No match found.').height(linehgt).width(elem.width()).  attr({'id': elemid + prefitem + 0})  .css(itemcssopt).bind('click', function(e) {  container.css('display', 'none');  });  container.append(tmprec); arrec.push(tmprec);  } else {  for (var i=0; i<data.length; i++) {  //debugger  var item = data[i];var tmpdisplay = item['display']; var tmpid = elemid + prefitem + i;  var tmprec = $('<div />').html(item['display']).height(linehgt).width(elem.width())  .attr('id', tmpid).css(itemcssopt)  .bind('mouseover', function(e){  $(this).css('background-color', hlColor);  })  .bind('mouseout', function(e){  $(this).css('background-color', unhlColor);  })  .bind('click', function(e) {  var tmpId = $(this).attr('id'); var tmpPrefIdx = tmpId.indexOf(prefitem);  var tmpFrag = tmpId.substring(tmpPrefIdx+prefitem.length, tmpId.length);  var tmpRecIdx = parseInt(tmpFrag); var tmpdata = ctrldata[tmpRecIdx];  elem.val(tmpdata['display']); container.css('display', 'none');  if (config.callback && (typeof config.callback=='function')) {  config.callback(tmpdata['value'], tmpdata['display']);  }  });  container.append(tmprec); arrec.push(tmprec);  }  }  if (elem) {  var pos = elem.position(); var newtop = pos.top + elem.height() + padding;  var totalheight = data.length>0 ? (data.length \* linehgt) : linehgt;  var actualheight = totalheight > maxheight ? maxheight : totalheight+5;  contentHeight = linehgt \* ctrldata.length;  container.width(elem.width()).height(actualheight).css(  {'display': 'inline', 'top': newtop, 'left': pos.left});  }  }  });  }, 2000);  break;  }  });    });  }    $.fn.autocomp.defaults = {chars: 3};  })(jQuery); |

# EXTEND OBJECTS WITH PROTOTYPE

**Function.prototype.call(thisArg[, arg1[, arg2[, ...]]])**

**Function.prototype.apply(thisArg[, argsArray])**

**PropUtil and describe**

|  |
| --- |
| <script>  //IE 8  if (!Object.create) {  Object.create = (function(){  function F(){}  return function(o){  if (arguments.length != 1) {  throw new Error('Object.create implementation only accepts one parameter.');  }  F.prototype = o;  return new F()  }  })()  }  function PropUtil() {  this.lvl = 0;  this.getPrototype = function(obj) {  return ( typeof obj.\_\_proto\_\_ === "object" ) ? obj.\_\_proto\_\_ : obj.constructor.prototype;  }  this.getVarName = function (v) {  for (var key in window) {  if (window[key] === v)  return key;  }  }  this.describe = function(obj, prev) {  if (typeof prev === 'undefined') this.lvl = 0;  if (typeof obj === 'undefined' || !obj) {  if (this.lvl>0) this.lvl--;  return;  }  if (this.lvl==0) document.write('<div>' + this.getVarName(obj) + '</div>');  //var proto = this.getPrototype(obj);  //if (typeof proto !== 'undefined' && proto && (typeof prev === 'undefined' || prev instanceof this.desfunc)) {  this.lvl++;  //document.write('<div style="text-indent:'+(20\*this.lvl)+'px;">' + '\_\_proto\_\_' + '</div>');  //this.describe(proto, obj);  for (var prop in obj) { //Object.keys(obj[prop]).length  if (prop) {  if (typeof obj[prop] !== 'undefined') {  if (this.getPrototype(obj[prop]) == Array.prototype) {  document.write('<div style="text-indent:'+(20\*this.lvl)+'px;">' + prop + ':' + obj[prop].join() + '</div>');  } else if (typeof obj[prop] === 'object') {  document.write('<div style="text-indent:'+(20\*this.lvl)+'px;">' + prop + ':' + '</div>');  this.describe(obj[prop], obj);  } else {  document.write('<div style="text-indent:'+(20\*this.lvl)+'px;">' + prop + ': ' + obj[prop] + '</div>');  }  }  }  }  //}  //this.lvl++;  }  this.implement = function(o, constructor){  // Another WebReflection Insane Snippet  if(o instanceof constructor)  // nothing to check  // classical boring stuff  return true;  // let's check if things are OK  var k, b = true,  // take the instance constructor prototype  po = o.constructor.prototype,  // take the implemented prototype  pc = constructor.prototype  ;  // for each property or method  // in the implemented constructor  for(k in pc)  // check if the instance inherited prototype  // has this method/property as well  b = b && k in po; //\* TOTHINK: \*/ && (typeof po[k] === typeof pc[k]); //\*/  // if there was nothing to check  // we cannot say a word ... but ...  // if instance has every method/property  // present in the compared constructor  // prototype, we could say this instance  // implements this constructor  return !!k && b;  };  }</script> |

Describe

|  |
| --- |
| var q = {  a: 2,  b: function(){  return Math.pow(this.a ,2);  },  c: {  d: "d",  e: function(f) {  return f + this.d;  }  }  };  var p = Object.create(q);  var pu = (new PropUtil())  pu.describe(p);  function Graph() {  this.vertices = [];  this.edges = [];  }  Graph.prototype = {  addVertex: function(v){  this.vertices.push(v);  }  };  var g = new Graph();  pu.describe(g); |

Describe result

|  |
| --- |
| p  a: 2  b: function (){ return Math.pow(this.a ,2); }  c:  d: d  e: function (f) { return f + this.d; }  g  vertices:  edges:  addVertex: function (v){ this.vertices.push(v); } |

# Inheritance thru prototype

|  |
| --- |
| <script>  function Mammal(name){  this.name=name;  this.offspring=[];  }  Mammal.prototype.haveABaby=function(){  var newBaby=new Mammal("Baby "+this.name);  this.offspring.push(newBaby);  return newBaby;  }  Mammal.prototype.toString=function(){  return '[Mammal "'+this.name+'"]';  }  Cat.prototype = new Mammal(); // Here's where the inheritance occurs  Cat.prototype.constructor=Cat; // Otherwise instances of Cat would have a constructor of Mammal  function Cat(name){  this.name=name;  }  Cat.prototype.toString=function(){  return '[Cat "'+this.name+'"]';  }  var someAnimal = new Mammal('Mr. Biggles');  var myPet = new Cat('Felix');  alert('someAnimal is '+someAnimal); // results in 'someAnimal is [Mammal "Mr. Biggles"]'  alert('myPet is '+myPet); // results in 'myPet is [Cat "Felix"]'  myPet.haveABaby(); // calls a method inherited from Mammal  alert(myPet.offspring.length); // shows that the cat has one baby now  alert(myPet.offspring[0]); // results in '[Mammal "Baby Felix"]'  //someAnimal is [Mammal "Mr. Biggles"]  //myPet is [Cat "Felix"]  //1  //[Mammal "Baby Felix"]  Cat.prototype.parent = Mammal.prototype; //super property  Cat.prototype.haveABaby=function(){  var theKitten = this.parent.haveABaby.call(this);  return theKitten;  }  //pure virtual function or interface  LivingThing = {  beBorn : function(){  this.alive=true;  }  }  ...  Mammal.prototype = LivingThing;  Mammal.prototype.parent = LivingThing; //Note: not 'LivingThing.prototype'  Mammal.prototype.haveABaby=function(){  this.parent.beBorn.call(this);  var newBaby=new this.constructor("Baby "+this.name);  this.offspring.push(newBaby);  return newBaby;  }  //new LivingThing(); will error out since its not a function  </script> |

# Convenient Inheritance

|  |
| --- |
| <script>  Function.prototype.inheritsFrom = function( parentClassOrObject ){  **if ( parentClassOrObject.constructor == Function )**  **{**  **//Normal Inheritance**  **this.prototype = new parentClassOrObject;**  **this.prototype.constructor = this;**  **this.prototype.parent = parentClassOrObject.prototype;**  **}**  **else**  **{**  **//Pure Virtual Inheritance**  **this.prototype = parentClassOrObject;**  **this.prototype.constructor = this;**  **this.prototype.parent = parentClassOrObject;**  **}**  **return this;**  }  //  //  LivingThing = {  beBorn : function(){  this.alive = true;  }  }  //  //  function Mammal(name){  this.name=name;  this.offspring=[];  }  Mammal.inheritsFrom( LivingThing );  Mammal.prototype.haveABaby=function(){  this.parent.beBorn.call(this);  var newBaby = new this.constructor( "Baby " + this.name );  this.offspring.push(newBaby);  return newBaby;  }  //  //  function Cat( name ){  this.name=name;  }  Cat.inheritsFrom( Mammal );  Cat.prototype.haveABaby=function(){  var theKitten = this.parent.haveABaby.call(this);  alert("mew!");  return theKitten;  }  Cat.prototype.toString=function(){  return '[Cat "'+this.name+'"]';  }  //  //  var felix = new Cat( "Felix" );  var kitten = felix.haveABaby( ); // mew!  alert( kitten ); // [Cat "Baby Felix"]  </script> |

# MIXIN

|  |
| --- |
| <script>  var asCircle = function() {  this.area = function() {  return Math.PI \* this.radius \* this.radius;  };  this.grow = function() {  this.radius++;  };  this.shrink = function() {  this.radius--;  };  return this;  };    var Circle = function(radius) {  this.radius = radius;  };  asCircle.call(Circle.prototype);  var circle1 = new Circle(5);  alert(circle1.area()); //78.54  </script> |

# CLASSIC COPY

|  |
| --- |
| <script>  var circleFns = {  area: function() {  return Math.PI \* this.radius \* this.radius;  },  grow: function() {  this.radius++;  },  shrink: function() {  this.radius--;  }  };  function extend(destination, source) {  for (var k in source) {  if (source.hasOwnProperty(k)) {  destination[k] = source[k];  }  }  return destination;  }  var RoundButton = function(radius, label) {  this.radius = radius;  this.label = label;  };    extend(RoundButton.prototype, circleFns);  var rbtn = new RoundButton(3, 'rbtn');  alert(rbtn.area());  new PropUtil().describe(rbtn);  </script> |

# EXTEND OBJECTS WITH JQUERY

|  |
| --- |
| Class.extend = function(prop /\*, prop, prop, prop \*/) {  for( var i=1, l=arguments.length; i<l; i++ ){  prop = $.extend( prop, arguments[i] );  }  } |

# INHERITING PARTS OF VARIOUS CLASSES

|  |
| --- |
| <script type="text/javascript" src="proputil.js"></script>  <script>  function A(){};  A.prototype.name = "A instance";  A.prototype.getName = function(){  return this.name;  };  function B(){};  B.prototype.name = "B instance";  function C(){};  // default name from B  C.prototype.name = B.prototype.name;  // getName from A  C.prototype.getName = A.prototype.getName;  // new method for C  C.prototype.setName = function(name){  this.name = name;  };  var c = new C();  c.getName(); // B instance  c.setName("Andrea");  c.getName(); // Andrea  new PropUtil().describe(c);  </script> |

# PRETTY PRINT

|  |
| --- |
| <script>  //pretty print  var tree = [  {item: 'red triangle', parent: 'triangle'},  {item: 'circle', parent: 'shape'},  {item: 'rectangle', parent: 'square'},  {item: 'filled circle', parent: 'circle'},  {item: 'blue triangle', parent: 'triangle'},  {item: 'shape', parent: null},  {item: 'empty circle', parent: 'circle'},  {item: 'square', parent: 'shape'},  {item: 'triangle', parent: 'shape'}  ];  function PrettyPrint() {  this.findParents = function(o) {  var out = [];  for (var i=0; i<o.length; i++) {  if (!o[i].parent) out.push(o[i]);  }  return out;  }  this.findChildren = function(o, p) {  var out = [];  for (var i=0; i<o.length; i++) {  if (o[i].parent==p.item) out.push(o[i]);  }  return out;  }  this.print = function(o) {  //find parents  var parents = this.findParents(o);  var lvl = 0;  for (var p=0; p<parents.length; p++) {  document.write('<div style="text-indent: '+(20\*lvl)+'px;">' + parents[p].item + '</div>');  this.\_print(o, lvl, parents[p]);  }  }  this.\_print = function(o, lvl, p) {  lvl++;  var children = this.findChildren(o, p);  for (var c=0; c<children.length; c++) {  var child = children[c];  document.write('<div style="text-indent: '+(20\*lvl)+'px;">' + child.item + '</div>');  this.\_print(o, lvl, child);  }  lvl--;  }  }  (new PrettyPrint()).print(tree);  </script> |

# TRAVERSE ALL DIVS AND HIGHLIGHT THEM WITH RED BORDER

|  |
| --- |
| <script type="text/javascript" src="jquery-1.11.1.js"></script>  <script>  //chrome.exe --disable-web-security  function fjq() {  $('#ifr').attr('src', $('#site').val()+'?output=embed');  $('#ifr').load(function(){  $('#ifr').contents().find($('#elem').val()).css('border', '1px #ff0000 solid');  });  }  function fdom() {  var ifr = document.getElementById('ifr');  var vsite = document.getElementById('site').value;  var vtag = document.getElementById('elem').value;  ifr.src = vsite;  ifr.onload = function() {  var tags = ifr.contentWindow.document.querySelectorAll(vtag);  for (var i=0; i<tags.length; i++) {  var t = tags[i];  if (typeof t.style !== 'undefined') {  t.style.borderWidth = '1px';  t.style.borderColor = '#ff0000';  t.style.borderStyle = 'solid';  }  }  }  }  </script>  site:<input type="text" name="site" id="site"/><br/>  elem:<input type="text" name="elem" id="elem"/><br/>  <button onclick="fdom();" name="js" id="js" style="float:left;">js</button>&nbsp;<button onclick="fjq();" name="jq" id="jq" style="float:left;">jq</button><br/>  <iframe name="ifr" id="ifr" width="1000" height="500" scroll="yes"></iframe> |

# NODE.JS

## Installations

node.js

python

PYTHONPATH=%PYTHONPATH%;path\_to\_python\_lib

PYTHONHOME=python\_home

Add %PYTHONHOME% to %PATH%

npm –install –g express

npm –install –g inspector

npm –install –g socket.io

mkdir /.npm-global

npm config set prefix '/.npm-global'

echo "export PATH=/.npm-global/bin:$PATH" >> ~/.profile

source ~/.profile

npm i -g @angular/cli

## NodeJs File IO

**var fs = require('fs');**

function copyData(savPath, srcPath) {

**fs.readFile(srcPath, 'utf8', function (err, data)** {

if (err) throw err;

//Do your processing, MD5, send a satellite to the moon, etc.

**fs.writeFile (savPath, data, function(err)** {

if (err) throw err;

console.log('complete');

});

});

}

function getFileContent(srcPath, callback) {

fs.readFile(srcPath, 'utf8', function (err, data) {

if (err) throw err;

callback(data);

}

);

}

function copyFileContent(savPath, srcPath) {

getFileContent(srcPath, function(data) {

fs.writeFile (savPath, data, function(err) {

if (err) throw err;

console.log('complete');

});

});

}

## http request

### request GET

var request = **require('request');**

// Set the headers

var headers = {

'User-Agent': 'Super Agent/0.0.1',

'Content-Type': 'application/x-www-form-urlencoded'

}

// Configure the request

var options = {

url: 'http://samwize.com',

method: 'GET',

headers: headers,

qs: {'key1': 'xxx', 'key2': 'yyy'}

}

// Start the request

**request(options, function (error, response, body)** {

if (!error && response.statusCode == 200) {

// Print out the response body

console.log(body)

}

})

### request POST

var request = require('request');

// Set the headers

var headers = {

'User-Agent': 'Super Agent/0.0.1',

'Content-Type': 'application/x-www-form-urlencoded'

}

//Configure the request

var options = {

url: 'http://samwize.com',

**method: 'POST'**,

headers: headers,

**form: {'key1': 'xxx', 'key2': 'yyy'}**

}

// Start the request

request(options, function (error, response, body) {

if (!error && response.statusCode == 200) {

// Print out the response body

console.log(body)

}

})

## uncompress gzip response

var uncompress = require('compress-buffer').uncompress;

// Set the headers and options ...

request(options, function (error, response, body) {

if (!error && response.statusCode == 200) {

// Handle gzip

var encoding = response.headers['content-encoding']

if(encoding && encoding.indexOf('gzip')>=0) {

body = uncompress(body);

}

body = body.toString('utf-8');

// Print out the response body

console.log(body)

// If it is json

var json\_body = JSON.parse(body);

}

})

## Express and serve html file

var express = require('express');

var app = express();

var path = require('path');

// viewed at http://localhost:8080

app.get('/', function(req, res) {

res.sendFile(path.join(\_\_dirname + '/index.html'));

});

app.listen(8080);

## Http module GET/POST

var http = require('http');

//The url we want is `www.nodejitsu.com:1337/`

var options = {

host: 'www.nodejitsu.com',

path: '/',

//since we are listening on a custom port, we need to specify it by hand

port: '1337',

//This is what changes the request to a POST request

method: **'POST' //’GET’**

};

callback = function(response) {

var str = ''

response.on('data', function (chunk) {

str += chunk;

});

response.on('end', function () {

console.log(str);

});

}

var req = http.request(options, callback);

//This is the data we are posting, it needs to be a string or a buffer

req.write("hello world!");

req.end();

//**http.get**(object | url-string , calback);