

# JAVASCRIPT VISUAL REFERENCE

## Variables



Variable names (labels) are case sensitive and can only start with a letter (A-Z, a-z), an underscore ( `_` ) or \$

```
var box2 = 24;    var name = true;
box3 = 24;        var box2 = null;
```

## Reserved keywords

<b>break</b>	<b>return</b>	<b>class</b>
<b>case</b>	<b>switch</b>	<b>const</b>
<b>catch</b>	<b>this</b>	<b>enum</b>
<b>continue</b>	<b>throw</b>	<b>export</b>
<b>debugger</b>	<b>try</b>	<b>extends</b>
<b>default</b>	<b>typeof</b>	<b>import</b>
<b>delete</b>	<b>var</b>	<b>super</b>
<b>do</b>	<b>void</b>	<b>implements</b>
<b>else</b>	<b>while</b>	<b>let</b>
<b>finally</b>		<b>private</b>
<b>for</b>		<b>public</b>
<b>function</b>		<b>yield</b>
<b>if</b>		<b>interface</b>
<b>in</b>		<b>package</b>
<b>instanceof</b>		<b>protected</b>
<b>new</b>		<b>static</b>
		<b>arguments</b>
		<b>eval</b>

## Bit operators

AND				XOR			
&	1	0		^	1	0	
	1	1	0		1	0	1
	0	0	0		0	1	0

OR				NOT (unary)			
	1	0		~	1	0	
	1	1	1		0	1	
	0	1	0				

## Operators

//Arithmetic Operators

```
2 * 3
4 / 2
4 + 3
4 - 3
4 % 3
```

//Assignment Operators

```
a = b = 2
a += 2
a *= 2
```

//Bitwise Operators

```
a & b;
a | b;
a ^ b;
```

//Comparison Operators

```
a === b;
a > b;
```

## Binary Operators

```
var a = 8;
console.log(a++); //8
```

```
var b = 8;
console.log(++b); //9
```

```
var a = true;
!a; //false
```

**a = -b;** //attempts to convert *b* to a number, if its not: returns NaN, //if it is, negates the value of *b* and assigns it to *a*. *b* retains it's value

**a = +b;** //attempts to convert *b* to a number, if it's not: returns NaN. *b* retains its value

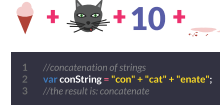
## Precedence



## String Escape Characters



cone + cat + ten + ate

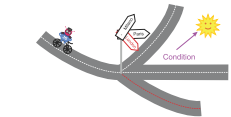


## String properties

**charAt()**  
**charCodeAt()**  
**concat()**  
**fromCharCode()**  
**indexOf()**  
**lastIndexOf()**  
**localeCompare()**  
**match()**  
**replace()**  
**search()**  
**slice()**  
**split()**  
**substr()**  
**substring()**  
**toLocaleLowerCase()**  
**toLocaleUpperCase()**  
**toLowerCase()**  
**toString()**  
**toUpperCase()**  
**trim()**  
**valueOf()**

Returns the character at the specified index (position)  
Returns the Unicode of the character at the specified index  
Joins two or more strings, and returns a copy of the joined strings  
Converts Unicode values to characters  
Returns the position of the first found occurrence of a specified value in a string  
Returns the position of the last found occurrence of a specified value in a string  
Compares two strings in the current locale  
Searches a string for a match against a regular expression, and returns the matches  
Searches a string for a value and returns a new string with the value replaced  
Searches a string for a value and returns the position of the match  
Extracts a part of a string and returns a new string  
Splits a string into an array of substrings  
Extracts a part of a string from a start position through a number of characters  
Extracts a part of a string between two specified positions  
Converts a string to lower case letters, according to the host's locale  
Converts a string to upper case letters, according to the host's locale  
Converts a string to lowercase letters  
Returns the value of a String object  
Converts a string to upper case letters  
Removes whitespace from both ends of a string  
Returns the primitive value of a String object

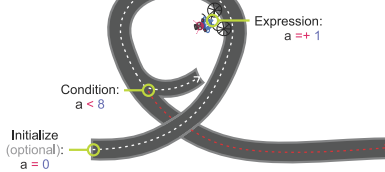
## Conditionals



```
if else else
var sunny = true;
rainy = false;
if (sunny) {
  console.log("keep straight");
} else if (sunny) {
  console.log("turn right");
} else {
  console.log("turn left");
}
```

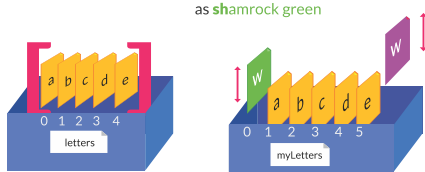
```
switch
var where = "London";
switch (where) {
  case "London":
    console.log("keep straight");
    break;
  case "Paris":
    console.log("keep straight");
    break;
  default:
    console.log("turn left");
}
```

## Loops



```
for
//add 1 to 'a' until 'a' is no longer less than 8
//initialize condition, expression
for (var a = 0; a < 8; a++) {
  alert(a);
}
//if 'a' is equal or greater than 8, go here
```

## Arrays



**concat()**  
**join()**  
**push()**  
**pop()**  
**reverse()**  
**shift()**  
**slice()**  
**sort()**  
**splice()**  
**unshift()**  
**delete()**

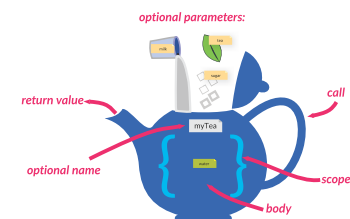
**concat()**  
**join()**  
add to the end  
remove and return the last item  
reverse the order of array items  
remove first item  
create new array from existing array items  
sort alphabetically  
replace items  
add at the beginning  
delete any element

## RE

**[abc]**  
**[0-9]**  
**(x|y|z)**  
**\d**  
**\s**  
**+**  
**^**  
**\$**  
**?**  
**.**

find any character between the brackets  
find any digit between the brackets  
find any of the separated with | sign  
find a digit  
find any word  
find one or more occurrences  
beginning of the string  
end of string  
zero or one  
find any character except newline

## Functions



```
var myTea = function (tea, sugar, milk) {
  var water = "Hot water.";
  return water + tea + "sugar" + "sugars" + "milk" + "milks";
};

//Call the function with custom parameters
myTea("green tea", 2, 3);
//Hot water, green tea, 2 sugars, 3 milks
```

## Function statement

```
function foo() { return 8; }
```

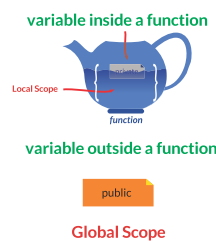
## Function expression

```
var foo = function foo() { return 8; };
var boo = function (x, y) { return x + y; };
(function doo() { return 8; })();
```

## IIFE

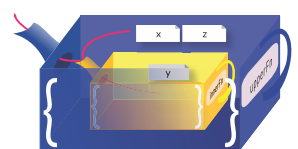
```
(function () {
  console.log("no need to call me");
})();
```

## Scope



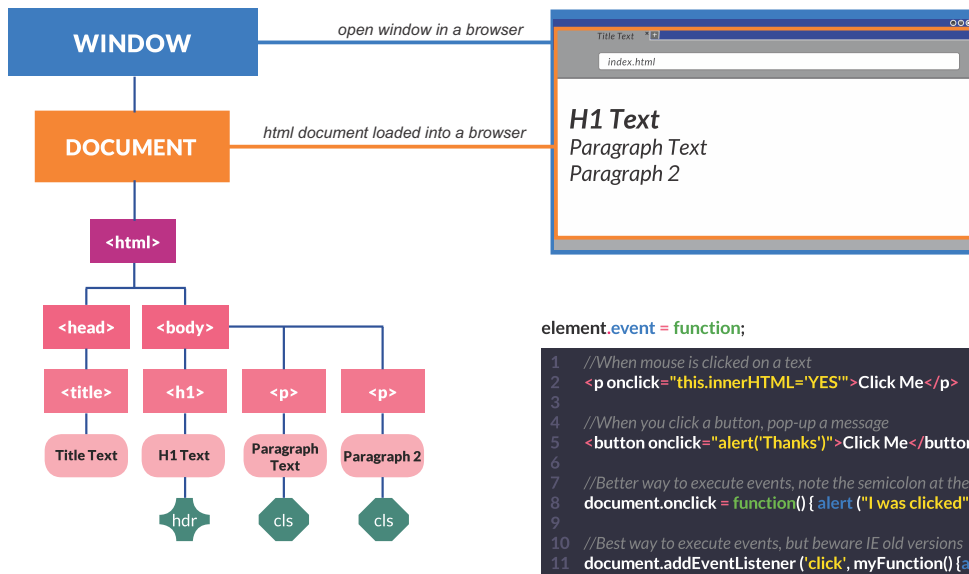
```
//Variable without the 'var' keyword
//is a global variable
function shadow() {
  public = "blue";
}
```

## Closure



```
function upperFn(x) {
  var z = 8; //local variable
  function innerFn(y) {
    console.log(x + y + z);
  }
  innerFn(3); //y = 3
}
upperFn(2); //x = 2
```

# DOM



index.html

```
1 <html>
2 <head>
3   <title>Title Text</title>
4 </head>
5 <body>
6   <h1 id="hdr">H1 Text</h1>
7   <p class="cls">Paragraph Text</p>
8   <p class="cls">Paragraph 2</p>
9   <script src="main.js"></script>
10 </body>
11 </html>
```

element.event = function;

```
1 //When mouse is clicked on a text
2 <p onclick="this.innerHTML='YES'">Click Me</p>
3
4 //When you click a button, pop-up a message
5 <button onclick="alert('Thanks')">Click Me</button>
6
7 //Better way to execute events, note the semicolon at the end
8 document.onclick = function() { alert("I was clicked"); };
9
10 //Best way to execute events, but beware IE old versions
11 document.addEventListener('click', myFunction() { alert("wow"); });
```

## WINDOW Properties

closed	returns boolean whether window is opened or not
defaultStatus	get/set property of default window status bar
document	reference to the current document object
frames	returns an array of all frames in the current window
history	returns URLs opened within the window
innerHeight	returns window's height in pixels
innerWidth	returns window's width in pixels
length	returns the number of frames in the current window
location	get/set the current URL
name	returns the name of the window
opener	reference to the window that opened a secondary window
outerHeight	returns entire window's height in pixels
outerWidth	returns entire window's width in pixels
parent	current window or frame set
screenLeft	X,Y coordinates of current window relative to user's screen
screenTop	
screenX	returns scroll's X location from upper left corner
screenY	
scrollX	returns scroll's Y location from upper left corner
scrollY	
self	current window
status	get/set the content of browser's status bar
top	topmost browser window

## DOCUMENT Properties

cookie  
documentMode  
domain  
lastModified  
readyState  
referrer  
title  
URL

## EVENT Handlers

onAbort	onMouseDown
onBlur	onMouseMove
onChange	onMouseOut
onClick	onMouseOver
onDbClick	onMouseUp
onDragDrop	onMove
onError	onReset
onFocus	onResize
onKeyDown	onSelect
onKeyPress	onSubmit
onKeyUp	onUnload
onLoad	

## WINDOW Methods

alert()	displays a dialog box with the specified message and OK button
blur()	removes focus from the window and brings it to the background
clearInterval()	clears a timer set with the setInterval() method
clearTimeout()	clears a timer set with the setTimeout() method
close()	closes a window
confirm()	displays a dialog box with the specified message and OK and Cancel buttons
focus()	sets focus to the window. Brings the window to the front
moveBy()	moves a window by the specified amount in pixels.
moveTo()	moves a window to the specified coordinates in pixels
open()	opens a new browser window
print()	prints the content of a window or frame
prompt()	displays a dialog box with a message and returns a string
resizeBy()	resizes a window by the specified amount in pixels
resizeTo()	resizes a window to the specified values in pixels
scrollBy()	scrolls a window by the specified amount in pixels
scrollTo()	scrolls a window to the specified values in pixels
setInterval()	calls a function or evaluates an expression at specified intervals (in milliseconds)
setTimeout()	calls a function or evaluates an expression after a specified time (in milliseconds)

## DOCUMENT Methods

close()  
open()  
getElementById()  
getElementsByClassName()  
getElementsByTagName()  
write()  
writeln()

```
1 //returns: all elements with the id "hdr"
2 getid = document.getElementById("hdr");
3 //<h1 id="hdr">H1 Text</h1>
4
5 //returns a node collection of all p tags
6 console.log(document.getElementsByTagName("p"));
7 //<p.cls, p.cls, item: function>
8
9 //returns all elements with the class name "cls"
10 console.log(document.getElementsByClassName("cls"));
11 //<p.cls, p.cls, item: function>
12
13 //Modify the content of the h1 element with the innerHTML property
14 getid.innerHTML = "Something Else";
15 //<h1 id="hdr">Something Else</h1>
```