CHEAT SHEET

JAVASCRIPT & AJAX FOR DUMMIES CHEAT SHEET

From JavaScript and AJAX For Dummies

By Andy Harris

The tables you find here offer a one-stop reference for the most common programming variables, commands, methods, and coding miscellany used in JavaScript programs, jQuery, and AJAX.

CODE TO USE IN JAVASCRIPT VARIABLE MANIPULATION FUNCTIONS

As shown in the following table, you can use these JavaScript statements in your own code to create and modify variables in your JavaScript functions.

Element	Description
var myVar = 0;	Creates a variable with given starting value. Type is determined dynamically.
stringVar = prompt("message")	Sends message to user in a dialog box, retrieves text input from user and stores it in stringVar.
stringVar.length	Returns the length (in characters) of stringVar.
stringVar.toUpperCase(), stringVar.toLowerCase()	Converts stringVar to upper- or lowercase.
stringVar.substring()	Returns a specified subset of stringVar.
stringVar.indexOf()	Returns location of a substring in stringVar (or -1).
parseInt()	Converts string to int.
parseFloat()	Converts string to float.
toString()	Converts any variable to string.
eval()	Evaluates string as JavaScript code.
Math.ceil()	Converts any number to integer by rounding up.
Math.floor()	Converts any number to integer by rounding down.
Math.round()	Converts any number to integer by standard rounding algorithm.
Math.random()	Returns random float between 0 and 1.

BASIC I/O COMMANDS IN JAVASCRIPT

JavaScript programmers commonly use the commands shown in the following table for controlling dialog-based input and output in programs to be used on the Web.

Element	Description
alert("message");	Creates a popup dialog containing "message."
stringVar = prompt("message")	Send message to user in a dialog box, retrieve text input from user and store it in stringVar.

JAVASCRIPT CONDITIONS AND BRANCHING CODE STRUCTURES

Look to the following table for JavaScript control structures you can use in your program code to add branching and looping behavior to your JavaScript programs.

Element	Description

if (condition){
// content
} else {
// more content
} // end if
	Executes content only if condition is true.
	Optional else clause occurs if condition
	is false.

switch (expression)
case: value;
//code
break;
default:
//code
}
	Compares expression against one or more values. If expression
	is equal to value, runs corresponding code.
	Default clause catches any uncaught values.

for(i = 0; i < count; i++)
//code
} // end for
	Repeats code i times.

While (condition){
//code
} // end while
	Repeats code as long as condition is true.

Function fnName(paramaters) {
//code
} // end function
	Defines a function named fnName and
	sends it parameters. All code inside the function will execute when
	the function is called.

ADD JAVASCRIPT COMPARISON OPERATORS TO CONDITION STATEMENTS

JavaScript uses comparison operators inside conditions to make numeric or alphabetical comparisons of variables to other variables or values. Using these operators, you can determine whether a variable is greater than, less than, or equal to another variable or value. You can also use combinations of these comparison operators.

Name	Operator	Example	Notes
Equality	==	(x==3)	Works with all variable types, including strings.
Not equal	!=	(x != 3)	True if values are not equal.
Less than	<	(x < 3)	Numeric or alphabetical comparison.
Greater than	>	(x > 3)	Numeric or alphabetical comparison.
Less than or equal to	<=	(x <= 3)	Numeric or alphabetical comparison.
Greater than or equal to	>=	(x >= 3)	Numeric or alphabetical comparison.

CREATE JAVASCRIPT STRUCTURES AND OBJECTS

JavaScript allows you to put together code lines to create functions and variables to create arrays. You can put functions and variables together to create objects.

Element	Description

function fnName(parameters) {
//code
} // end function
	Defines a function named fnName and
	sends it parameters. All code inside function will execute when the
	function is called.

var myArray = new Array("a", "b", "c");	Creates an array. Elements can be any type (even mixed types).

Var myJSON = {
"name":
"Andy",
"title":
"Author"
}
	Creates a JSON object. Each element
	has a name/value pair, and can contain anything, including an array
	(with square braces) another JSON object,
	or a function.

Var person = new Object(); Person.name = "Andy";	Creates an object. You can add ordinary variables (which become properties) or functions (which become methods).

CHANGE YOUR WEB PAGE WITH JAVASCRIPT DOCUMENT OBJECT MODEL METHODS

The Document Object Model methods shown in the following list offer you a great way to access and modify your Web pages through your JavaScript code.

- **myElement = document.getElementById(“name“);:** Gets an element from the page with the specified ID and copies a reference to that element to the variable myElement.
- **myElement.innerHTML = “value“:** Changes the value of the element to “value“.
- **document.onkeydown = keyListener:** When a key is pressed, a function called keyListener is automatically activated.

- **document.onmousemove = mouseListener:** When the mouse is moved, a function called mouseListener is automatically activated.

- **setInterval(function, ms);:** Runs function each ms milliseconds.
- **myArray = document.getElementByName("name"):** Returns an array of objects with the current name (frequently used with radio buttons).

ADD SEARCHING TOOLS WITH REGULAR EXPRESSION OPERATORS IN JAVASCRIPT

The regular expression mechanism adds extremely powerful searching tools to your programming. Here are some of the most commonly used regular expressions as they are used in JavaScript.

Operator and Description	Sample pattern	Matches and Doesn't match
. (period) Any single character except newline	.	E n
^ Beginning of string	^a	apple Banana
$ End of string	a$	banana Apple
[characters] Any of a list of characters in braces	[abcABC]	A D
[char range] Any character in the range	[a-zA-Z]	F 9
d Any single numerical digit	ddd-dddd	123-4567 The-thing
b A word boundary	btheb	the Theater
+ One or more occurrences of the previous character	d+	1234 Text
* Zero or more occurrences of the previous character	[a-zA-Z]d*	B17, g 7
{digit} Repeat preceding character *digit* times	d{3}-d{4}	123-4567 999-99-9999
{min, max} Repeat preceding character at least *min* but not more than *max* times	.{2,4}	Ca, com, info watermelon
(pattern segment) Store results in pattern memory returned with code	^(.).*1$	gig, wallow Bobby

COMMON METHODS OF THE JQUERY NODE

The jQuery library turns DOM objects into powerful jQuery nodes. The following table shows a few of the more commonly used methods of the jQuery node.

Method	Description
addClass(), removeClass(), toggleClass()	Applies or removes a CSS class to a jQuery node.
css("attribute", "value")	Applies a single CSS rule to the jQuery node.
Css(JSONObject)	Applies JSON object list of CSS rules and values to the jQuery node.
html()	Reads or changes the HTML contents of the jQuery node.
text()	Reads or changes the text contents of a jQuery node.
val()	Reads the value of a form element.
bind(event, function)	Triggers function to occur when event occurs.
Show(), hide(), toggle()	Makes element appear or disappear.
animate(parameters, duration)	parameters is a JSON object consisting of CSS rules and values. Values are smoothly changed from current value to target value over duration (measured in milliseconds).

JQUERY SELECTORS AND FILTERS

Part of jQuery's power is based on its ability to select particular sections of the page. This table includes several commonly used selectors and filters.

Selector/Filter	Searches for...
$("element")	Any HTML element.
$("#elementID")	Any element with the given ID.
$(".className")	Any element with the given class name.
:header	Any header tag (h1, h2, h3, and so on).
:animated	Any element that is currently being animated.
:contains(text)	Any element that contains the indicated text.
:empty	The element is empty.
:parent	An element that contains some other element.
:attribute=value	The element has an attribute with the specified value.
:Input, :text, :radio, :image, :button, etc	Matches on the specific element type (especially useful for form elements that are all variations of the input tag).

ADD JQUERY USER INTERFACE CLASSES TO THEME STYLES

These CSS classes are defined in a jQuery UI theme. If you're using jQuery UI, you can add any of these classes to your objects to add the theme styles.

Class	Used on	Description
ui-widget	Outer container of widget	Makes element look like a widget.
ui-widget-header	Heading element	Applies distinctive heading appearance.
ui-widget-content	Widget	Applies widget content style to element and children.
ui-state-default	Clickable elements	Displays standard (unclicked) state.
ui-state-hover	Clickable elements	Displays hover state.
ui-state-focus	Clickable elements	Display focus state when element has keyboard focus.
ui-state-active	Clickable elements	Display active state when mouse is clicked on element.
ui-state-highlight	Any widget or element	Specifies element is currently highlighted.
ui-state-error	Any widget or element	Specifies an element will contain an error or warning message.
ui-state-error text	Text element	Allows error highlighting without changing other elements (mainly used in form validation).
ui-state-disabled	Any widget or element	Demonstrates that widget is currently disabled.
ui-corner-all, ui-corner-tl (etc)	Any widget or element	Adds current corner size to element. Specify specific corners with tl, tr, bl, br, top, bottom, left, right.
ui-widget-shadow	Any widget	Applies shadow effect to widget.

JQUERY METHODS FOR SENDING AN AJAX REQUEST

As you can see from studying the following table, jQuery supplies several methods for sending an AJAX request to the server and parsing the results.

Method	Description
get(url, parameters)	Send an HTTP GET request to the given URL. Parameters is JSON object encapsulating form data (name/value pairs). Result is returned as HTML, XML, or plain text data.
post(url, parameters)	Just like get, but uses the post method, which hides the parameters.
load(url, parameters)	Much like get(), but returns a jQuery object. Calling jQuery objects contents are replaced by the returned data (usually HTML or XHTML).
getJSON	Like get, but returns a JSON object, which can be parsed for further processing.