

**HTML**



# THE MEGA CHEAT SHEET

**TAGS - EVENT ATTRIBUTES -  
MOBILE - BROWSERS - CANVAS**

**by Jamie Spencer @**

make a  **websitehub.com**

# HTML 5 TAGS

Below is a full list of all the HTML 5 Tags / elements and a description of what they are used for.

## NEW HTML 5 TAGS

<b>&lt;article&gt;</b>	self-contained composition that is independently distributable	<b>&lt;menuitem&gt;</b>	specifies a command that a user can invoke from a popup menu
<b>&lt;aside&gt;</b>	section of page that consists of content tangentially related to content around it	<b>&lt;mark&gt;</b>	marked text
<b>&lt;audio&gt;</b>	sound content	<b>&lt;meter&gt;</b>	measurement in defined range
<b>&lt;bdi&gt;</b>	span of text to be isolated from surroundings for bidirectional formatting purposes	<b>&lt;nav&gt;</b>	navigation links
<b>&lt;canvas&gt;</b>	area that can be used to draw graphics via JavaScript	<b>&lt;output&gt;</b>	represents results of calculation
<b>&lt;command&gt;</b>	user invokable command	<b>&lt;progress&gt;</b>	progress of any kind of task
<b>&lt;datalist&gt;</b>	dropdown list	<b>&lt;rb&gt;</b>	marks the base text component of a ruby annotation.
<b>&lt;datatemplate&gt;</b>	data template	<b>&lt;rp&gt;</b>	parenthesized ruby text
<b>&lt;details&gt;</b>	details of an element	<b>&lt;rt&gt;</b>	ruby text
<b>&lt;dialog&gt;</b>	specifies that part of an application is interactive	<b>&lt;rtc&gt;</b>	marks a ruby text container for ruby text components in a ruby annotation
<b>&lt;embed&gt;</b>	embedded content	<b>&lt;ruby&gt;</b>	ruby annotations
<b>&lt;figcaption&gt;</b>	caption of figure element	<b>&lt;section&gt;</b>	section in a document
<b>&lt;figure&gt;</b>	group of media content	<b>&lt;source&gt;</b>	media resources
<b>&lt;footer&gt;</b>	footer for section or page	<b>&lt;summary&gt;</b>	header of a detail element
<b>&lt;header&gt;</b>	header for section or page	<b>&lt;template&gt;</b>	declares HTML fragments that can be cloned and inserted in the document by script.
<b>&lt;hgroup&gt;</b>	group of headings for section	<b>&lt;time&gt;</b>	date/time
<b>&lt;keygen&gt;</b>	generated key in a form	<b>&lt;track&gt;</b>	specifies a text track for media such as video and audio
<b>&lt;main&gt;</b>	specifies the main content area of an HTML document	<b>&lt;video&gt;</b>	video
		<b>&lt;wbr&gt;</b>	possible line break

## OLD UNSUPPORTED TAGS

<b>&lt;acronym&gt;</b>	acronym	<b>&lt;isindex&gt;</b>	provides searchable index related to current document
<b>&lt;applet&gt;</b>	applet	<b>&lt;dir&gt;</b>	directory list
<b>&lt;basefont&gt;</b>	base font	<b>&lt;noembed&gt;</b>	no embed section
<b>&lt;bgsound&gt;</b>	background sound	<b>&lt;noframes&gt;</b>	no frame section
<b>&lt;big&gt;</b>	big text	<b>&lt;s&gt;</b>	strikethrough text
<b>&lt;center&gt;</b>	centered text	<b>&lt;strike&gt;</b>	strikethrough text
<b>&lt;fn&gt;</b>	footnotes	<b>&lt;tt&gt;</b>	teletype text
<b>&lt;font&gt;</b>	text font, size, and color	<b>&lt;u&gt;</b>	underlined text
<b>&lt;frame&gt;</b>	sub window	<b>&lt;xmp&gt;</b>	preformatted text
<b>&lt;frameset&gt;</b>	set of frames		

## EXISTING HTML 5 TAGS

<code>&lt;!--...--&gt;</code>	comment	<code>&lt;label&gt;</code>	label for a form control
<code>&lt;!doctype&gt;</code>	document type	<code>&lt;legend&gt;</code>	title in a fieldset
<code>&lt;a&gt;</code>	hyperlink	<code>&lt;li&gt;</code>	list item
<code>&lt;abbr&gt;</code>	abbreviation	<code>&lt;link&gt;</code>	resource reference
<code>&lt;address&gt;</code>	address element	<code>&lt;map&gt;</code>	image map
<code>&lt;area&gt;</code>	Image map area	<code>&lt;menu&gt;</code>	menu list
<code>&lt;b&gt;</code>	bold text	<code>&lt;meta&gt;</code>	meta information
<code>&lt;base&gt;</code>	base URL for all links in page relative to document root	<code>&lt;noscript&gt;</code>	no script section
<code>&lt;bdo&gt;</code>	text direction	<code>&lt;object&gt;</code>	embedded object
<code>&lt;blockquote&gt;</code>	long quotation	<code>&lt;ol&gt;</code>	ordered list
<code>&lt;body&gt;</code>	body element	<code>&lt;optgroup&gt;</code>	option group
<code>&lt;br&gt;</code>	single line break	<code>&lt;option&gt;</code>	option in a drop-down list
<code>&lt;button&gt;</code>	push button	<code>&lt;p&gt;</code>	paragraph
<code>&lt;caption&gt;</code>	table caption	<code>&lt;param&gt;</code>	parameter for an object
<code>&lt;cite&gt;</code>	citation	<code>&lt;pre&gt;</code>	preformatted object
<code>&lt;code&gt;</code>	code text	<code>&lt;q&gt;</code>	short quotation
<code>&lt;col&gt;</code>	attributes for columns	<code>&lt;samp&gt;</code>	sample computer code
<code>&lt;colgroup&gt;</code>	groups of columns	<code>&lt;script&gt;</code>	script
<code>&lt;dd&gt;</code>	definition of description	<code>&lt;select&gt;</code>	selectable list
<code>&lt;del&gt;</code>	deleted text	<code>&lt;small&gt;</code>	small text
<code>&lt;div&gt;</code>	generic block-level element	<code>&lt;span&gt;</code>	inline generic container
<code>&lt;dfn&gt;</code>	defining instance of a term	<code>&lt;strong&gt;</code>	strong text
<code>&lt;dl&gt;</code>	definition list	<code>&lt;style&gt;</code>	style definition
<code>&lt;dt&gt;</code>	definition term	<code>&lt;sub&gt;</code>	subscripted text
<code>&lt;em&gt;</code>	emphasized text	<code>&lt;sup&gt;</code>	superscripted text
<code>&lt;fieldset&gt;</code>	logically group items in a form	<code>&lt;table&gt;</code>	table
<code>&lt;form&gt;</code>	defines a form	<code>&lt;tbody&gt;</code>	table body
<code>&lt;h1&gt; to &lt;h6&gt;</code>	header 1 to header 6	<code>&lt;td&gt;</code>	table cell
<code>&lt;head&gt;</code>	document information	<code>&lt;textarea&gt;</code>	text area
<code>&lt;hr&gt;</code>	horizontal rule	<code>&lt;tfoot&gt;</code>	table footer
<code>&lt;html&gt;</code>	html document	<code>&lt;th&gt;</code>	table header
<code>&lt;i&gt;</code>	italic text	<code>&lt;thead&gt;</code>	wraps row containing table headers
<code>&lt;iframe&gt;</code>	inline sub window	<code>&lt;title&gt;</code>	document title
<code>&lt;img&gt;</code>	image	<code>&lt;tr&gt;</code>	table row
<code>&lt;input&gt;</code>	input field	<code>&lt;ul&gt;</code>	unordered list
<code>&lt;ins&gt;</code>	inserted text	<code>&lt;var&gt;</code>	variable
<code>&lt;kbd&gt;</code>	keyboard text		

## Existing Attributes In HTML4 & 5

A list of existing attributes that are support in both HTML4 & HTML 5

<b>onabort</b>	abort event	<b>onmouseout</b>	mouse cursor exited event
<b>onbeforeunload</b>	before unload event	<b>onmouseover</b>	mouse cursor entered event
<b>onclick</b>	mouse clicked event	<b>onmouseup</b>	mouse button released event
<b>ondblclick</b>	mouse double-clicked event	<b>onmove</b>	move event
<b>ongesturechange</b>	gesture change event	<b>onorientationchange</b>	device orientation changed event
<b>ongestureend</b>	end of gesture event	<b>onpaste</b>	paste event
<b>ongesturestart</b>	start of gesture event	<b>onresize</b>	resize event
<b>onkeydown</b>	key pressed down event	<b>ontouchcancel</b>	touch canceled event
<b>onkeypress</b>	key pressed event	<b>ontouchend</b>	touch ended event
<b>onkeyup</b>	key released event	<b>ontouchmove</b>	touch move event
<b>onmousedown</b>	mouse button pressed down event	<b>ontouchstart</b>	touch started event
<b>onmousemove</b>	mouse moved event		













# HTML 5 Browser Support













Below is a table showing the individual browser support specifications for each of the major web browsers in both desktop and mobile. We have given an over all percentage score on each browser, based on scores gathered by [html5test.com](http://html5test.com)

● Fully Supported

● Partially Supported

● Not Supported

	Details & Summary Elements	WebGL - 3D Canvas Graphics	Inline SVG in HTML5	Datalist Element	Progress & Meter	ClassList (DOM TokenList)	Ruby Annotation	Form Validation	Session History Mgmt	HTML5 Form Features	Drag & Drop
<b>Desktop</b>											
11.0 	●	●	●	●	●	●	●	●	●	●	●
38.0 	●	●	●	●	●	●	●	●	●	●	●
43.0 	●	●	●	●	●	●	●	●	●	●	●
8.0 	●	●	●	●	●	●	●	●	●	●	●
15.0 	●	●	●	●	●	●	●	●	●	●	●
<b>Mobile</b>											
2.3 	●	●	●	●	●	●	●	●	●	●	●
8.0 	●	●	●	●	●	●	●	●	●	●	●
29.0 	●	●	●	●	●	●	●	●	●	●	●
7.0 	●	●	●	●	●	●	●	●	●	●	●
35.0 	●	●	●	●	●	●	●	●	●	●	●
43.0 	●	●	●	●	●	●	●	●	●	●	●
43.0 	●	●	●	●	●	●	●	●	●	●	●

		Content Editable Attribute	Text API for Canvas	Audio Element	Video Element	Data & Data (Attributes)	New Semantic Elements	Canvas (Basic Support)	Hashchange Event	Offline Web Apps	Get Elements by Class Name	
<b>Desktop</b>												
11.0		●	●	●	●	●	●	●	●	●	●	61%
38.0		●	●	●	●	●	●	●	●	●	●	83%
43.0		●	●	●	●	●	●	●	●	●	●	95%
8.0		●	●	●	●	●	●	●	●	●	●	71%
29.0		●	●	●	●	●	●	●	●	●	●	94%
<b>Mobile</b>												
5.0		●	●	●	●	●	●	●	●	●	●	81%
6.0		●	●	●	●	●	●	●	●	●	●	73%
12.0		●	●	●	●	●	●	●	●	●	●	88%
7.0		●	●	●	●	●	●	●	●	●	●	41%
35.0		●	●	●	●	●	●	●	●	●	●	82%
43.0		●	●	●	●	●	●	●	●	●	●	89%
8.1		●	●	●	●	●	●	●	●	●	●	62%

## Overall Support of HTML 5 On Desktop Browsers



61%



83%



95%



71%



94%

## Overall Support of HTML 5 On Mobile Browsers



81%



73%



88%



41%



82%



89%



62%

## Event Handler Content Attributes

HTML 4 added the ability to let events trigger actions in a browser, like starting a JavaScript when a user clicks on an element.

Below are event handler attributes that can be added to HTML elements to define event actions.

<b>onbeforeonload</b>	before onload event	<b>onloadeddata</b>	media data is loaded
<b>oncanplay</b>	media can start play	<b>onloadedmetadata</b>	duration of media element is loaded
<b>oncanplaythrough</b>	media can be played to end	<b>onloadstart</b>	browser starts to load media data
<b>oncontextmenu</b>	context menu is triggered	<b>onmessage</b>	element is invalid
<b>ondrag</b>	element is dragged	<b>onmousewheel</b>	mouse wheel is being rotated
<b>ondragend</b>	at the end of drag operation	<b>onpause</b>	media data is paused
<b>ondragenter</b>	element dragged on drop target	<b>onplay</b>	media data is going to start playing
<b>ondragleave</b>	element leaves valid drop target	<b>onplaying</b>	media data has started playing
<b>ondragover</b>	element is dragged over drop target	<b>onprogress</b>	browser is fetching media data
<b>ondragstart</b>	at the start of drag operation	<b>onratechange</b>	media data's pplaying rate has changed
<b>ondrop</b>	dragged element is being dropped	<b>onscroll</b>	element's scrollbar is being scrolled
<b>ondurationchange</b>	length of media is changed	<b>onseeked</b>	element's seeking attribute is not true
<b>onemptied</b>	media resource element becomes empty	<b>onseeking</b>	element's seeking attribute is true
<b>onended</b>	media has reached end	<b>onstalled</b>	there is error in fetching media data
<b>onerror</b>	error occurs	<b>onstorage</b>	document loads
<b>onformchange</b>	form changes	<b>onsuspend</b>	browser has stopped fetching media data
<b>onforminput</b>	form gets user input	<b>ontimeupdate</b>	media changes its playing position
<b>onhaschange</b>	document has changed	<b>onvolumechange</b>	media changes volume, also when mute
<b>oninput</b>	message is triggered	<b>onwaiting</b>	media has stopped playing
<b>oninvalid</b>	element gets user input		



# HTML 5 Canvas

The HTML <canvas> element is used to draw graphics, on the fly, via scripting (usually JavaScript). The <canvas> element is only a container for graphics. You must use a script to actually draw the graphics.

Canvas has several methods for drawing paths, boxes, circles, text, and adding images.

## CANVAS ELEMENT

### ATTRIBUTES

Name	Type	Default
Width	Unsigned Long	300
Height	Unsigned Long	150

### METHODS

Return	Name
String	toDataURL([Optional] string type, [Variadic] any args)
Object	getContext9 string contextId)

## 2D CONTEXT

### ATTRIBUTES

Name	Type
Canvas	HTMLCanvasObject [readonly]

### METHODS

Return	Name
Void	save()
Void	restore()

## TRANSFORMATION

### METHODS

Return	Name
Void	scale(float x, float y)
Void	rotate (float x, float y)
Void	transform(float m11, float m12, float m21, float m22, float dx, float dy)
Void	setTransform(float m11, float m12, float m21, float m22, float dx, float dy)

## IMAGE DRAWING

### METHODS

Return	Name
Void	drawImage(Object image, float dx, float dy, [Optional] float dw, float dh)
Argument "image" can be of type HTMLImageElement, HTMLCanvasElement or HTMLVideoElement	
Void	drawImage(float sx, float sy, float sw, float sh, float dx, float dy, float dw, float dh)

## COLORS, STYLES AND SHADOWS

### ATTRIBUTES

Name	Type	Default
strokeStyle	any	black
fillStyle	any	black
shadowOffsetX	float	0.0
shadowOffsetY	float	0.0
shadowBlur	float	0.0
shadowColor	string	transparent black

### METHODS

Return	Name
CanvasGradient	createLinearGradient(float x0, float y0, float r0, float x1, float y1, float r1)
CanvasPattern	createPattern(Object image, string repetition)

Argument "image" can be of type HTMLImageElement, HTMLCanvasElement or HTMLVideoElement  
"repetition" supports any of the following values:  
[repeat (default), repeat-x, repeat-y, no-repeat]

### CANVASGRADIENT INTERFACE

Return	Name
Void	addColorStop(float offset, string color)

# HTML 5 Canvas

The HTML <canvas> element is used to draw graphics, on the fly, via scripting (usually JavaScript). The <canvas> element is only a container for graphics. You must use a script to actually draw the graphics.

Canvas has several methods for drawing paths, boxes, circles, text, and adding images.

## PATHS

### METHODS

Return	Name
Void	beginPath( )
Void	closePath( )
Void	fill( )
Void	stroke( )
Void	clip( )
Void	moveTo( float x, float y)
Void	lineTo( float x, float y)
Void	quadraticCurveTo( float cpx, float cpy, float x, float y)
Void	bezierCurveTo( float cp1x, float cp1y, float cp2x, float cp2y, float x, float y)
Void	arcTo( float x1, float y1, float x2, float y2, float radius )
Void	arc( float x1, float y1, float radius ) float startAngle, float endAngle, boolean anticlockwise )
Void	rect( float x, float w, float h) isPointInPath( float x, float y)

## PIXEL MANIPULATION

### METHODS

Return	Name
ImageData	createImageData( float sw, float sh)
ImageData	createImageData( ImageData)
ImageData	getImageData( ImageData imagedata, float dx, float dy, [Optional] float dirtyX, float dirtyY, float dirtyWidth, float dirtyHeight)

### ImageData interface

width	unsigned long	[readonly]
height	unsigned long	[readonly]
data	CanvasPixelArray	[readonly]

## TEXT

### ATTRIBUTES

Name	Type	Default
font	String	10px sans-serif
textAlign	String	start

Supports any of the following values:  
[start, end, left, right, center]

textBaseline	String	alphabetic
--------------	--------	------------

Supports an of the following values:  
[top, hanging, middle, alphabetic, ideographic, bottom]

### METHODS

Return	Name
Void	fillText( string text, float x, float y, [Optional] float maxWidth)
Void	strokeText( string text, float x, float y, [Optional] float maxWidth)
TextMetrics	measureText( string text)

### TextMetrics interface

width	float [readonly]
-------	------------------

## RECTANGLES

### METHODS

Return	Name
Void	clearRect( float x, float y, float w, float h)
Void	fillRect( float x, float y, float w, float h)
Void	strokeRect( float x, float y, float w, float h)

# HTML 5 Canvas

The HTML <canvas> element is used to draw graphics, on the fly, via scripting (usually JavaScript). The <canvas> element is only a container for graphics. You must use a script to actually draw the graphics.

Canvas has several methods for drawing paths, boxes, circles, text, and adding images.

## CANVAS ELEMENT

### ATTRIBUTES

Name	Type	Default
globalAlpha	Float	1
globalCompositeOperation	String	Source-over

Supports any of the following values:



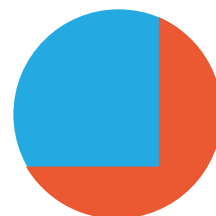
source-over



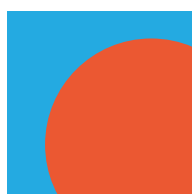
source-in



destination-out



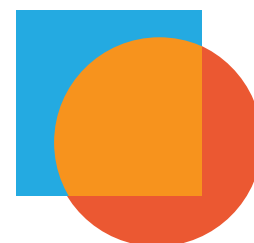
destination-atop



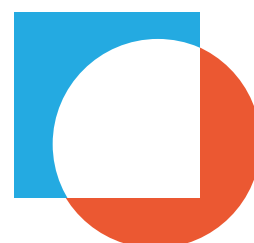
source-atop



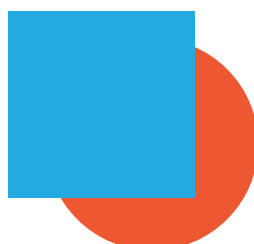
source-out



lighter



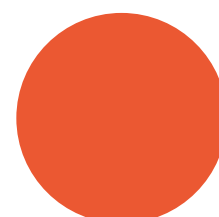
XOR



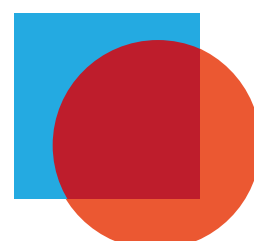
destination-atop



destination-in



copy



darker

# HTML 5 Canvas

The HTML <canvas> element is used to draw graphics, on the fly, via scripting (usually JavaScript). The <canvas> element is only a container for graphics. You must use a script to actually draw the graphics.

Canvas has several methods for drawing paths, boxes, circles, text, and adding images.

## LINE STYLES

### ATTRIBUTES

Name	Type	Default
lineWidth	Float	1
lineCap	String	Butt

Supports any of the following values:

butt



rounded



square



lineJoin

String

miter

Supports any of the following values:

round



bevel



miter



miterLimit

Float

10