

HTML



THE MEGA CHEAT SHEET

TAGS - EVENT ATTRIBUTES - MOBILE - BROWSERS - CANVAS

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

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

OLD UNSUPPORTED TAGS

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

EXISTING HTML 5 TAGS

<!--...-->	comment	<label>	label for a form control
<!doctype>	document type	<legend>	title in a fieldset
<a>	hyperlink		list item
<abbr>	abbreviation	<link>	resource reference
<address>	address element	<map>	image map
<area>	Image map area	<menu>	menu list
	bold text	<meta>	meta information
<base>	base URL for all links in page relative to document root	<noscript>	no script section
<bdo>	text direction	<object>	embedded object
<blockquote>	long quotation		ordered list
<body>	body element	<optgroup>	option group

	single line break	<option>	option in a drop-down list
<button>	push button	<p>	paragraph
<caption>	table caption	<param>	parameter for an object
<cite>	citation	<pre>	preformatted object
<code>	code text	<q>	short quotation
<col>	attributes for columns	<samp>	sample computer code
<colgroup>	groups of columns	<script>	script
<dd>	definition of description	<select>	selectable list
	deleted text	<small>	small text
<div>	generic block-level element		inline generic container
<dfn>	defining instance of a term		strong text
<dl>	definition list	<style>	style definition
<dt>	definition term	<sub>	subscripted text
	emphasized text	<sup>	superscripted text
<fieldset>	logically group items in a form	<table>	table
<form>	defines a form	<tbody>	table body
<h1> to <h6>	header 1 to header 6	<td>	table cell
<head>	document information	<textarea>	text area
<hr>	horizontal rule	<tfoot>	table footer
<html>	html document	<th>	table header
<i>	italic text	<thead>	wraps row containing table headers
<iframe>	inline sub window	<title>	document title
	image	<tr>	table row
<input>	input field		unordered list
<ins>	inserted text	<var>	variable
<kbd>	keyboard text		

Existing Attributes In HTML4 & 5

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

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

● Not Supported

[illegible]

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.

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

CANVAS ELEMENT

ATTRIBUTES

Name	Type	Default
globalAlpha	Float	1
globalCompositeOperation	String	Source-over

Supports any of the following values:



source-over



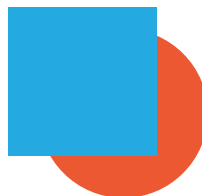
source-in



source-atop



source-out



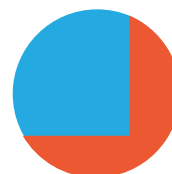
destination-atop



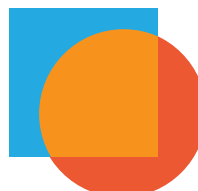
destination-in



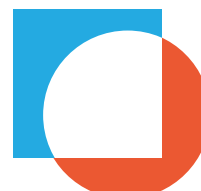
destination-out



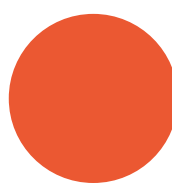
destination-atop



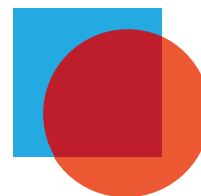
lighter



XOR



copy



darker

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)

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]

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

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)