

# HTML5 Cheat Sheet

HTML5, CSS3, Geolocation und Pinned Sites auf einen Blick!

## Vereinfachungen

`<!DOCTYPE html>` Neuer HTML5-Doctype.

`<html lang="de"></html>` HTML-Tag mit Sprachangabe ohne XML-Namensraum.

`<meta charset="utf-8" />` Angabe der Zeichenkodierung.

`<link rel="stylesheet" href="Url" />` Einbinden eines Stylesheets ohne Angabe des CSS MIME-Typs.

`<script></script>`, `<style></style>` Einbinden von JavaScript-Code bzw. eines CSS-Stylesheets ohne Angabe des MIME-Typs.

## Semantische Tags

`<section>` Spezifiziert einen beliebigen Teilbereich eines Dokuments.

`<nav>` Bereich einer Seite, der Verweise zu anderen Seiten enthält.

`<article>` Spezifiziert eine abgeschlossene und inhaltlich eigenständige Abfassung (z.B. ein Artikel, ein Kommentar, etc.) innerhalb einer Seite.

`<aside>` Spezifiziert einen Bereich, der keinen oder nur einen geringen Bezug zu dem ihn umgebenden Dokument hat. (z.B. eine Worterklärung, ein Zitat, etc.)

`<hgroup>` Repräsentiert die Überschrift einer Seite, eines Artikels oder einer Sektion. Kann neben einer Hauptüberschrift auch beliebige Unter-Überschriften besitzen.

`<header>` Kopfbereich einer Seite oder eines darin eingeschlossenen Teildokuments. Umfasst üblicherweise Überschriften, Logos oder Einleitungstexte.

`<footer>` Fußbereich einer Seite oder eines darin eingeschlossenen Teildokuments. Enthält üblicherweise Fußnoten, Quellenangaben, Wortklärungen oder Rechtstexte.

`<time datetime="JJJ-MM-TTThh:mm:ss-hh:mm" pubdate> Text</time>` Spezifiziert eine Uhrzeit und/oder ein Datum (z.B. 2011-12-24 oder 2011-12-24T20:30:00+01:00). Das Attribut `pubdate` gibt an, ob es sich bei dem angegebenen Zeitstempel um das Veröffentlichungsdatum des umgebenden `<article>`-Elements bzw. der gesamten Seite handelt.

`<mark>` Eine hervorgehobene Textpassage.

## `<video>` / `<audio>`

Spielt eine Video- bzw. Audio-Datei ab. Unterstützte Formate variieren nach Browser.

### ATTRIBUTE

**src="Url"** URL der abzuspielenden Video- oder Audio-Datei.

**width="Pixel"** Breite des Players in Pixeln (nur `<video>`).

**height="Pixel"** Höhe des Players in Pixeln (nur `<video>`).

**poster="Url"** Url eines Platzhalter-Bildes (nur `<video>`).

**autoplay** Wiedergabe automatisch starten.

**controls** Steuerelemente anzeigen.

**muted** Stummschaltung aktivieren.

**loop** Wiedergabe automatisch wiederholen.

**preload="Wert"** Video-/Audio-Datei beim Öffnen der Seite laden. Mögliche Werte: *none* (nichts vorladen), *metadata* (nur Metadaten), *auto* (automatisch).

### `<source>`

Spezifiziert die Video- oder Audio-Datenquelle eines `<video>` oder `<audio>`-Elements. Kann mehrfach vorkommen, wenn für das Element kein `src`-Attribut gesetzt ist.

### ATTRIBUTE

**src="Url"** URL der abzuspielenden Video- oder Audio-Datei.

**type="MIME-Typ"** MIME-Typ und Codec der abzuspielenden Datei, z.B. `video/mp4`; `codecs="avc1.42E01E, mp4a.40.2"` oder `audio/mp3`

## HTMLVideoElement / HTMLAudioElement

Zu `<audio>` bzw. `<video>` Tag korrespondierender DOM-Elementtyp

### ATTRIBUTE UND METHODEN

**play()** Wiedergabe starten.

**pause()** Wiedergabe anhalten.

**currentTime** Aktuelle Wiedergabeposition in Sekunden (lesen/schreiben).

**duration** Länge der Medien-Datei in Sekunden.

**canPlayType(MIME-Typ)** Liefert einen Wert zurück, der angibt, ob der spezifizierte Medientyp abgespielt werden kann. Mögliche Werte:  
„“ (Leer) Kann nicht abgespielt werden  
*maybe* Möglicherweise (je nach Codec)  
*probably* Wahrscheinlich

### `<canvas>`

Erzeugt eine zweidimensionale Zeichenfläche, die per JavaScript angesteuert werden kann.

### ATTRIBUTE

**width="Pixel"** Breite der Zeichenfläche in Pixeln.

**height="Pixel"** Höhe der Zeichenfläche in Pixeln.



## HTMLCanvasElement

Zu `<canvas>` Tag korrespondierender DOM-Elementtyp.

### ATTRIBUTE UND METHODEN

**getContext("2d")** Liefert den 2D-Zeichenkontext des Canvas-Elements zurück.

## CanvasRenderingContext2D

2D-Zeichenkontext des Canvas-Elements.

### ATTRIBUTE

**strokeStyle** Strich-Stil, z.B. `red`, `#FF14EA`, `rgb(128,0,13)`

**fillStyle** Füll-Stil, wie `strokeStyle`

**font** Schriftart, z.B. `bold 14px Arial`

**textAlign** Horizontale Text-Ausrichtung. Mögliche Werte: `start`, `end`, `left`, `right`, `center`.

**textBaseline** Text-Grundlinie. Mögliche Werte: `top`, `hanging`, `middle`, `alphabetic`, `ideographic`, `bottom`.

**lineWidth** Strichstärke.

**globalAlpha** Globaler Transparenz-Wert (0 = transparent, 1 = opak).

**globalCompositeOperation** Globaler Überzeichnungseffekt.

Mögliche Werte: `source-over`, `source-in`, `source-out`, `source-atop`, `destination-over`, `destination-in`, `destination-out`, `destination-atop`, `lighter`, `copy`, `xor`.

**shadowBlur** Weichzeichnen-Radius des Schattens in Pixeln.

**shadowColor** Schattenfarbe, z.B. `#000000`

**shadowOffsetX**, **shadowOffsetY** Horizontaler und vertikaler Schatten-Versatz in Pixeln

### METHODEN

**drawImage(Bild, X, Y, Breite<sub>opt</sub>, Höhe<sub>opt</sub>)** Zeichnet ein Bild an die angegebene Position mit der angegebenen Breite und Höhe. Das Quellbild kann ein Image, Canvas oder Video-Objekt sein.

**drawImage(Bild, X<sub>1</sub>, Y<sub>1</sub>, Breite<sub>1</sub>, Höhe<sub>1</sub>, X<sub>2</sub>, Y<sub>2</sub>, Breite<sub>2</sub>, Höhe<sub>2</sub>)** Zeichnet einen Bildausschnitt (X<sub>1</sub>, Y<sub>1</sub>, Breite<sub>1</sub>, Höhe<sub>1</sub>) an die angegebene Position (X<sub>2</sub>, Y<sub>2</sub>, Breite<sub>2</sub>, Höhe<sub>2</sub>).

**beginPath()** Startet einen neuen Zeichenpfad.

**closePath()** Schließt den aktuellen Zeichenpfad.

**isPointInPath(X, Y)** Gibt true zurück, wenn der angegebene Punkt innerhalb des aktuellen Zeichenpfads liegt, sonst false.

**fill()** Füllt die aktuelle Form mit dem gesetzten fillStyle.

**stroke()** Füllt die Kontur der aktuellen Form mit dem gesetzten strokeStyle.

**moveTo(X, Y)** Bewegt Zeichenstift an die angegebene Position.

**lineTo(X, Y)** Zeichnet eine Linie von der letzten zur angegebenen Position.

**quadraticCurveTo(K<sub>x</sub>, K<sub>y</sub>, X, Y)** Zeichnet eine quadratische Bézier-Kurve von der letzten zur angegebenen Position (X, Y) unter Verwendung eines Kontrollpunkts (K<sub>x</sub>, K<sub>y</sub>).

**bezierCurveTo(K<sub>x1</sub>, K<sub>y1</sub>, K<sub>x2</sub>, K<sub>y2</sub>, X, Y)** Zeichnet eine kubische Bézier-Kurve von der letzten zur angegebenen Position (X, Y) unter Verwendung zweier Kontrollpunkte (K<sub>x1</sub>, K<sub>y1</sub> und K<sub>x2</sub>, K<sub>y2</sub>).

**arcTo(X<sub>1</sub>, Y<sub>1</sub>, X<sub>2</sub>, Y<sub>2</sub>, Radius)** Zeichnet ein Kreissegment mit angegebenem Radius von der letzten Position unter Verwendung der angegebenen Kontrollpunkte (X<sub>1</sub>, Y<sub>1</sub>, X<sub>2</sub>, Y<sub>2</sub>).

**arc(X, Y, Radius, StartWinkel, EndWinkel, Richtung)** Zeichnet ein Kreissegment mit dem angegebenen Mittelpunkt (X, Y), Radius und Anfangs- und Endwinkel in festgelegter Richtung (false = im Uhrzeigersinn, true = gegen den Uhrzeigersinn).

**rect(X, Y, Breite, Höhe)** Zeichnet ein Rechteck mit angegebener Position und Ausmaßen.

**fillText(Text, X, Y, MaxBreite<sub>opt</sub>)** Zeichnet den angegebenen Text ausgefüllt an die spezifizierte Position unter Berücksichtigung der maximalen Breite und unter Verwendung des gesetzten fillStyle.

**strokeText(Text, X, Y, MaxBreite<sub>opt</sub>)** Wie fillText, allerdings wird der Text unter Verwendung des strokeStyle-Attributs konturiert gezeichnet.

**measureText(Text)** Liefert ein TextMetrics-Objekt zurück, dessen width-Attribut die Breite des angegebenen Texts in Pixeln zurückliefert.

**clearRect(X, Y, Breite, Höhe)** Löscht den angegebenen Bereich des Canvas und macht ihn transparent.

**fillRect(X, Y, Breite, Höhe)** Zeichnet ein ausgefülltes Rechteck an die angegebene Position mit den spezifizierten Ausmaßen unter Verwendung des fillStyle.

**strokeRect(X, Y, Breite, Höhe)** Wie fillRect, zeichnet jedoch nur die Rechtecks-Kontur und verwendet strokeStyle.

**getImageData(X, Y, Breite, Höhe)** Liefert ein ImageData-Objekt für das angegebene Quell-Rechteck zurück. Das Attribut data des Objekts enthält die Pixel-Informationen in Form eines Arrays, wobei jeweils vier aufeinanderfolgende Werte die Farbinformation (Rot, Grün, Blau, Alpha) eines Pixels enthalten.

**createImageData(Breite, Höhe)** Erzeugt ein neues ImageData-Objekt mit angegebener Breite und Höhe (siehe getImageData).

**putImageData(ImageData, X, Y)** Zeichnet das angegebene ImageData-Objekt an die spezifizierte Stelle.

**save()** Sichert die aktuelle Transformations-Matrix und weitere Zustandsinformationen auf einem Stack.

**restore()** Stellt den zuletzt mit save gesicherten Zustand wieder her.

**scale(X, Y)** Skaliert die aktuelle Transformations-Matrix um die angegebenen Werte.

**rotate(Winkel)** Rotiert die aktuelle Transformations-Matrix um den angegebenen Winkel im Bogenmaß (Bogenmaß = Grad \* Math.PI / 180).

**translate(X, Y)** Verschiebt die aktuelle Transformations-Matrix um die angegebenen Werte.

**transform(Werte)** Multipliziert die aktuelle Transformations-Matrix mit der angegebenen 3x3 Matrix (nur 6 Werte erforderlich).

**setTransform(Werte)** Ersetzt die aktuelle Transformations-Matrix durch die angegebene 3x3 Matrix (nur 6 Werte erforderlich).

**createLinearGradient(Start<sub>x</sub>, Start<sub>y</sub>, End<sub>x</sub>, End<sub>y</sub>),**

**createRadialGradient(Start<sub>x</sub>, Start<sub>y</sub>, Radius, End<sub>x</sub>, End<sub>y</sub>)** Erzeugt ein Objekt, das einen linearen bzw. radialen Farbverlauf mit den angegebenen Koordinaten beschreibt. Kann z.B. dem Attribut fillStyle oder strokeStyle zugewiesen werden. Das Objekt verfügt über eine Methode **addColorStop(Versatz, Farbe)**, über das sich die einzelnen Farben des Verlaufs festlegen lassen.

**createPattern(Bild, Wiederholung)** Erzeugt ein Bildmuster mit der angegebenen Wiederholung. Mögliche Werte für Wiederholung: repeat, repeat-x, repeat-y, no-repeat. Kann z.B. dem Attribut fillStyle oder strokeStyle zugewiesen werden.

## CSS3

**@font-face { font-family: Schriftname; src: url(Url); }** Eigene Schriftart festlegen. Die Schriftdatei muss im WOFF-Format unter der angegebenen URL abgelegt werden.

Anwendung: font-family: Schriftname;

**border-radius: Radius;** bzw. **border-radius: Horizontal Vertikal;** Versieht das Zielelement mit abgerundeten Ecken.

**border-\*-\*-radius: Radius;** bzw.

**border-\*-\*-radius: Horizontal Vertikal;** Spezifiziert den Radius einzeln für alle vier Ecken, z.B. border-top-left-radius: 10px;

**box-shadow: inset Horizontversatz Vertikalversatz Unschärfegrad Ausbreitung Farbe;** Versieht das Zielelement mit einem Schlagschatten. Der Wert inset gibt an, dass der Schatten innerhalb des Elements gezeichnet werden soll.

**-ms-transform-origin: X Y;** Spezifiziert die Ursprungskordinaten für die folgenden Transformationen. Der Präfix -ms- ist Microsoft-spezifisch. Weitere Präfixe anderer Browserhersteller sind -webkit-, -moz- und -o-.

**-ms-transform: Funktion(Parameter);** Transformiert das Zielelement mit der angegebenen Funktion. Mehrfachangaben durch Hintereinander-Reihung sind möglich (Leerzeichen-getrennt).

translate(X, Y) Verschiebung in X- und Y-Richtung

translateX(Wert), translateY(Wert) Verschiebung in X- bzw. Y-Richtung

scale(X, Y) Skalieren in X- und Y-Richtung

scaleX(Wert), scaleY(Wert) Skalieren in X- bzw. Y-Richtung

rotate(Winkel) Drehen um angegebenen Winkel, z.B. 45deg

skew(X, Y) Neigen in X- und Y-Richtung

skewX(Wert), skewY(Wert) Neigen in X- bzw. Y-Richtung

matrix(Werte) Mit 3x3 Matrix transformieren (nur 6 Werte erforderlich)



**opacity: Wert;** Legt die Transparenz des Zielelements fest (0 = transparent, 1 = opak).

## Geolocation

Ermöglicht das Bestimmen der geographischen Position des Geräts, über das die Website aufgerufen wird.

**navigator.geolocation.getCurrentPosition(Callback);** Ermittelt die aktuelle Position und ruft die spezifizierte Callback-Funktion einmalig auf. Die Callback-Funktion hat die Signatur: function(position) {}, wobei position ein Objekt mit den folgenden Attributen ist:

coords Objekt mit Positionsdaten. Enthält u.A. die Attribute latitude (Breitengrad) und longitude (Längengrad).

timestamp Zeit, zu der die Position ermittelt wurde.

**var watchId = navigator.geolocation.watchPosition(Callback);** Wie getCurrentPosition, allerdings wird die Callback-Funktion bei jeder Positionsänderung erneut aufgerufen.

**navigator.geolocation.clearWatch(WatchId);** Verhindert weitere Aufrufe einer über watchPosition registrierten Callback-Funktion. Die zu übergebende WatchId wird von watchPosition zurückgeliefert.

## Pinned Sites

Ermöglicht das Anheften einer Website an die Windows Task-Leiste.

**<link rel="shortcut icon" href="Uri" />** Legt das Symbol fest, das in der Task-Leiste angezeigt werden soll. Hierbei muss es sich um eine Datei im ICO-Format handeln.

**<meta name="Name" content="Wert" />** Legt grundlegende Eigenschaften der angehefteten Seite fest:

application-name Name der Seite, erscheint beim Rollover über das Symbol

msapplication-tooltip Kurzbeschreibung der Seite

msapplication-starturl URL, die bei Klick auf das Symbol geöffnet wird

msapplication-window Ausmaße des Browser-Fensters, z.B. width=1024;height=768

msapplication-navbutton-color Farbe des Vor- und Zurück-Buttons

**<meta name="msapplication-task" content="Parameter" />** Ermöglicht das Hinzufügen von Aufgaben zur Jump-Liste der Seite (erscheint bei Rechtsklick auf das Symbol). Parameter müssen wie folgt angegeben werden: name=Titel;action-uri=Url;icon-uri=Url

## Links

**INTERNET EXPLORER HERUNTERLADEN**  
[www.internet-explorer.de](http://www.internet-explorer.de)

**MSDN-SEITEN ZU HTML5**  
<http://msdn.de/ie/aa740476>

**HTML5 DEMOS UND BENCHMARKS**  
<http://ie.microsoft.com/testdrive/>