BEGINNER'S

HTML CHEAT SHEET

WebsiteSetup

Main root	**2**
Document metadata	**2**
Sectioning root	**3**
Content sectioning	**3**
Text content	**4**
Inline text semantics	**6**
Image and multimedia	**8**
Scripting	**9**
Demarcating edits	**9**
Table content	**9**
Forms	**11**
Interactive elements	**12**

Main root

<html> … </html>

The HTML <html> element represents the root (top-level element) of an HTML document, so it is also referred to as the root element. All other elements must be descendants of this element.

Example:

```
<!DOCTYPE html>
<html lang="en">
  <head>...</head>
  <body>...</body>
</html>
```

Document metadata

<head> … </head>

The HTML <head> element contains machine-readable information (metadata) about the document, like its title, scripts, and style sheets.

<link>

The HTML External Resource Link element (<link>) specifies relationships between the current document and an external resource. This element is most commonly used to link to stylesheets, but is also used to establish site icons (both "favicon" style icons and icons for the home screen and apps on mobile devices) among other things.

<meta>

The HTML <meta> element represents metadata that cannot be represented by other HTML meta-related elements, like <base>, <link>, <script>, <style> or <title>

<style> … </style>

The HTML <style> element contains style information for a document, or part of a document.

<title> … </title>

The HTML Title element (<title>) defines the document's title that is shown in a browser's title bar or a page's tab.

Example:

```
<!DOCTYPE html>
<html lang="en">
  <head>...</head>
  <body>...</body>
</html>
```

Sectioning root

<body> … </body>

The HTML <body> Element represents the content of an HTML document. There can be only one <body> element in a document.

Example:

```
<html>
  <head>
    <title>Document title</title>
  </head>
  <body>
    <p>This is a paragraph</p>
  </body>
</html>
```

Content sectioning

<address> … </address>

The HTML <address> element indicates that the enclosed HTML provides contact information for a person or people, or for an organization.

<article> … </article>

The HTML <article> element represents a self-contained composition in a document, page, application, or site, which is intended to be independently distributable or reusable (e.g., in syndication).

<aside> … </aside>

The HTML <aside> element represents a portion of a document whose content is only indirectly related to the document's main content.

<footer> … </footer>

The HTML <footer> element represents a footer for its nearest sectioning content or sectioning root element. A footer typically contains information about the author of the section, copyright data or links to related documents.

<header> … </header>
The HTML <header> element represents introductory content, typically a group of introductory or navigational aids. It may contain some heading elements but also a logo, a search form, an author name, and other elements.

<h1> to <h6> … </h6>
The HTML <h1>–<h6> elements represent six levels of section headings. <h1> is the highest section level and <h6> is the lowest.

<main> … </main>
The HTML <main> element represents the dominant content of the <body> of a document. The main content area consists of content that is directly related to or expands upon the central topic of a document, or the central functionality of an application.

<nav> … </nav>
The HTML <nav> element represents a section of a page whose purpose is to provide navigation links, either within the current document or to other documents. Common examples of navigation sections are menus, tables of contents, and indexes.

<section> … </section>
The HTML <section> element represents a standalone section — which doesn't have a more specific semantic element to represent it — contained within an HTML document.

Example:

```
<address>
  <a href="mailto:jim@rock.com">jim@rock.com</a><br>
  <a href="tel:+13115552368">(311) 555-2368</a>
</address>
```

Text content

<blockquote> … </blockquote>
The HTML <blockquote> Element (or HTML Block Quotation Element) indicates that the enclosed text is an extended quotation. Usually, this is rendered visually by indentation (see Notes for how to change it). A URL for the source of the quotation may be given using the cite attribute, while a text representation of the source can be given using the <cite> element.

<dd> … </dd>
The HTML <dd> element provides the description, definition, or value for the preceding term (<dt>) in a description list (<dl>).

** … **
The HTML Content Division element ( ) is the generic container for flow content. It has no effect on the content or layout until styled using CSS.

<dl> … </dl>
The HTML <dl> element represents a description list. The element encloses a list of groups of terms (specified using the <dt> element) and descriptions (provided by <dd> elements). Common uses for this element are to implement a glossary or to display metadata (a list of key-value pairs).

<dt> … </dt>
The HTML <dt> element specifies a term in a description or definition list, and as such must be used inside a <dl> element.

<figcaption> … </figcaption>
The HTML <figcaption> or Figure Caption element represents a caption or legend describing the rest of the contents of its parent <figure> element.

<figure> … </figure>
The HTML <figure> (Figure With Optional Caption) element represents self-contained content, potentially with an optional caption, which is specified using the (<figcaption>) element.

<hr>
The HTML <hr> element represents a thematic break between paragraph-level elements: for example, a change of scene in a story, or a shift of topic within a section.

** … **
The HTML <li> element is used to represent an item in a list.

** … **
The HTML <ol> element represents an ordered list of items, typically rendered as a numbered list.

** … **
The HTML element represents a paragraph.

<pre> … </pre>
The HTML <pre> element represents preformatted text which is to be presented exactly as written in the HTML file.

** … **
The HTML <ul> element represents an unordered list of items, typically rendered as a bulleted list.

Examples:

```
<dl>
    <dt>Denim (semigloss finish)</dt>
    <dd>Ceiling</dd>

    <dt>Denim (eggshell finish)</dt>
    <dt>Evening Sky (eggshell finish)</dt>
    <dd>Layered on the walls</dd>
</dl>

<figure>
    <img src="/media/examples/elephant-660-480.jpg"
         alt="Elephant at sunset">
    <figcaption>An elephant at sunset</figcaption>
</figure>

<ol>
  <li>Mix flour, baking powder, sugar, and salt.</li>
  <li>In another bowl, mix eggs, milk, and oil.</li>
  <li>Stir both mixtures together.</li>
  <li>Fill muffin tray 3/4 full.</li>
  <li>Bake for 20 minutes.</li>
</ol>
```

Inline text semantics

**<a> … **

The HTML <a> element (or anchor element), with its href attribute, creates a hyperlink to web pages, files, email addresses, locations in the same page, or anything else a URL can address.

<abbr> … </abbr>

The HTML Abbreviation element (<abbr>) represents an abbreviation or acronym; the optional title attribute can provide an expansion or description for the abbreviation.

**
**

The HTML
 element produces a line break in text (carriage-return). It is useful for writing a poem or an address, where the division of lines is significant.

<cite> … </cite>

The HTML Citation element (<cite>) is used to describe a reference to a cited creative work, and must include the title of that work.

<code> … </code>

The HTML <code> element displays its contents styled in a fashion intended to indicate that the text is a short fragment of computer code.

…

The HTML element marks text that has stress emphasis. The element can be nested, with each level of nesting indicating a greater degree of emphasis.

<mark> … </mark>

The HTML Mark Text element (<mark>) represents text which is marked or highlighted for reference or notation purposes, due to the marked passage's relevance or importance in the enclosing context.

<small> … </small>

The HTML <small> element represents side-comments and small print, like copyright and legal text, independent of its styled presentation. By default, it renders text within it one font-size small, such as from small to x-small.

…

The HTML element is a generic inline container for phrasing content, which does not inherently represent anything. It can be used to group elements for styling purposes (using the class or id attributes), or because they share attribute values, such as lang.

…

The HTML Strong Importance Element () indicates that its contents have strong importance, seriousness, or urgency. Browsers typically render the contents in bold type.

<time> … </time>

The HTML <time> element represents a specific period in time.

Examples:

```
<blockquote>
    <p>It was a bright cold day in April, and the clocks were striking
thirteen.</p>
    <footer>
        First sentence in <cite><a
href="http://www.george-orwell.org/1984/0.html"><em>Nineteen
Eighty-Four</em></a></cite> by George Orwell (Part 1, Chapter 1).
    </footer>
</blockquote>
<p>You can use <abbr title="Cascading Style Sheets">CSS</abbr> to style your
<abbr title="HyperText Markup Language">HTML</abbr>.</p>
```

Image and multimedia

<audio> … </audio>

The HTML <audio> element is used to embed sound content in documents. It may contain one or more audio sources, represented using the src attribute or the <source> element: the browser will choose the most suitable one. It can also be the destination for streamed media, using a MediaStream.

<img>

The HTML <img> element embeds an image into the document.

<track>

The HTML <track> element is used as a child of the media elements <audio> and <video>. It lets you specify timed text tracks (or time-based data), for example to automatically handle subtitles. The tracks are formatted in WebVTT format (.vtt files) — Web Video Text Tracks or Timed Text Markup Language (TTML).

<video> … </video>

The HTML Video element (<video>) embeds a media player which supports video playback into the document. You can use <video> for audio content as well, but the <audio> element may provide a more appropriate user experience.

Examples:

```
<img class="fit-picture"
     src="/media/examples/grapefruit-slice-332-332.jpg"
     alt="Grapefruit slice atop a pile of other slices">

<figure>
    <figcaption>Listen to the T-Rex:</figcaption>
    <audio
        controls
        src="/media/examples/t-rex-roar.mp3">
            Your browser does not support the
            <code>audio</code> element.
    </audio>
</figure>

<video controls width="250"
       src="/media/examples/friday.mp4">

    <track default kind="captions"
           srclang="en"
```

```
            src="/media/examples/friday.vtt"/>

    Sorry, your browser doesn't support embedded videos.
</video>
```

Scripting

<script> ... </script>
The HTML <script> element is used to embed or reference executable code; this is typically used to embed or refer to JavaScript code.

Example:

```
<!-- HTML4 -->
<script type="text/javascript" src="javascript.js"></script>

<!-- HTML5 -->
<script src="javascript.js"></script>
```

Demarcating edits

<del> … </del>
The HTML <del> element represents a range of text that has been deleted from a document.

<ins> … </ins>
The HTML <ins> element represents a range of text that has been added to a document.

Example:

```
<p>"You're late!"</p>
<del>
    <p>"I apologize for the delay."</p>
</del>
<ins cite="../howtobeawizard.html" datetime="2018-05">
    <p>"A wizard is never late …"</p>
</ins>
```

Table content

<caption> … </caption>
The HTML Table Caption element (<caption>) specifies the caption (or title) of a table, and if used is always the first child of a <table>.

<table> … </table>
The HTML <table> element represents tabular data — that is, information presented in a two-dimensional table comprised of rows and columns of cells containing data.

<tbody> … </tbody>
The HTML Table Body element (<tbody>) encapsulates a set of table rows (<tr> elements), indicating that they comprise the body of the table (<table>).

<td> … </td>
The HTML <td> element defines a cell of a table that contains data. It participates in the table model.

<tfoot> … </tfoot>
The HTML <tfoot> element defines a set of rows summarizing the columns of the table.

<th> … </th>
The HTML <th> element defines a cell as header of a group of table cells. The exact nature of this group is defined by the scope and headers attributes.

<thead> … </thead>
The HTML <thead> element defines a set of rows defining the head of the columns of the table.

<tr> … </tr>
The HTML <tr> element defines a row of cells in a table. The row's cells can then be established using a mix of <td> (data cell) and <th> (header cell) elements.

Example:

```
<table>
    <thead>
        <tr>
            <th colspan="2">The table header</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>The table body</td>
            <td>with two columns</td>
        </tr>
    </tbody>
</table>
```

Forms

<button> … </button>
The HTML <button> element represents a clickable button, which can be used in forms or anywhere in a document that needs simple, standard button functionality.

<datalist> … </datalist>
The HTML <datalist> element contains a set of <option> elements that represent the values available for other controls.

<fieldset> … </fieldset>
The HTML <fieldset> element is used to group several controls as well as labels (<label>) within a web form.

<form> … </form>
The HTML <form> element represents a document section that contains interactive controls for submitting information to a web server.

<input>
The HTML <input> element is used to create interactive controls for web-based forms in order to accept data from the user; a wide variety of types of input data and control widgets are available, depending on the device and user agent.

<label> … </label>
The HTML <label> element represents a caption for an item in a user interface.

<legend> … </legend>
The HTML <legend> element represents a caption for the content of its parent <fieldset>.

<optgroup> … </optgroup>
The HTML <optgroup> element creates a grouping of options within a <select> element.

<option> … </option>
The HTML <option> element is used to define an item contained in a <select>, an <optgroup>, or a <datalist> element. As such, <option> can represent menu items in popups and other lists of items in an HTML document.

<progress> … </progress>
The HTML <progress> element displays an indicator showing the completion progress of a task, typically displayed as a progress bar.

`<select> … </select>`
The HTML <select> element represents a control that provides a menu of options

`<textarea> … </textarea>`
The HTML <textarea> element represents a multi-line plain-text editing control, useful when you want to allow users to enter a sizeable amount of free-form text, for example a comment on a review or feedback form.

Example:

```
<form action="" method="get" class="form-example">
  <div class="form-example">
    <label for="name">Enter your name: </label>
    <input type="text" name="name" id="name" required>
  </div>
  <div class="form-example">
    <label for="email">Enter your email: </label>
    <input type="email" name="email" id="email" required>
  </div>
  <div class="form-example">
    <input type="submit" value="Subscribe!">
  </div>
</form>
```

Interactive elements

`<details> … </details>`
The HTML Details Element (<details>) creates a disclosure widget in which information is visible only when the widget is toggled into an "open" state.

`<summary> … </summary>`
The HTML Disclosure Summary element (<summary>) element specifies a summary, caption, or legend for a <details> element's disclosure box.

Example:

```
<details>
    <summary>Details</summary>
    Something small enough to escape casual notice.
</details>
```

Source: https://developer.mozilla.org/en-US/docs/Web/HTML/Element

Beginner's Essential

CSS CHEAT SHEET

What makes a website unique is its styling. A must-have skill for every web developer.

WebsiteSetup

Table of Contents

Backgrounds	3
Border	4
Box Model	7
Font	9
Text	10
Column	12
Colors	13
Table	13
Speech	14
List & Markers	16
Animations	17
Transitions	18
UI	18
Pseudo-Class	20
Pseudo-Element	21
Absolute Measurement	21
Relative Measurement	21
Angles	22
Time	22
Frequency	22
Colors	23
Selector Types	23
Outline	25

3D / 2D Transform	25
Generated Content	26
Line Box	28
Hyperlink	31
Positioning	31
Ruby	32
Paged Media	32

Backgrounds

background

```
background-image

background-position

background-size

background-repeat

background-attachment

background-origin

background-clip

background-color
```

background-image

```
url

gradients

none
```

background-position

```
top left | top center | top right | center left | center center |
center right | bottom left | bottom center | bottom right

x-% y-%

x-pos y-pos
```

background-size

```
length

%

auto | cover | contain
```

background-repeat

```
repeat | repeat-x | repeat-y |

no-repeat
```

background-attachment

scroll | fixed | local

background-origin

border-box | padding-box | content-box

background-clip

border-box | padding-box | content-box

background-color

color

transparent

Border

border

border-width

border-style

border-color

border-width

thin | medium | thick | length

border-style

none | hidden | dotted |

dashed | solid | double |

groove | ridge | inset | outset

border-color

color

border-bottom

border-bottom-width

border-style

border-color

border-left

border-left-width

border-style

border-color

border-left-style

border-style

border-right-color

border-color

border-right-width

thin | medium | thick | length

border-top-width

thin | medium | thick | length

border-break

border-width

border-style

color

close

border-bottom-color

border-color

border-bottom-style

border-style

border-left-color

border-color

border-left-width

thin | medium | thick length

border-right-style

border-style

border-top

border-top-width

border-style

border-color

border-top-color

```
border-color
```

border-top-style

```
border-style
```

box-shadow

```
inset || [ length, length, length, length || <color> ]

none
```

border-collapse

```
collapse | separate
```

border-image

```
image

[ number / % border-width stretch | repeat | round ]

none
```

border-right

```
border-right-width

border-style

border-color
```

border-radius

```
border-radius

border-top-right-radius

border-bottom-right-radius

border-bottom-left-radius

border-top-left-radius
```

border-top-right-radius

```
length
```

border-bottom-right-radius

```
length
```

border-bottom-left-radius

```
length
```

Box Model

float

```
left | right | none
```

height

```
auto

length

%
```

max-height

```
none

length

%
```

max-width

```
none

length%
```

min-height

```
none

length

%
```

width

```
auto

%

length
```

margin

```
margin-top

margin-right

margin-bottom

margin-left
```

margin-bottom

auto

length

%

margin-left

auto

height

%

margin-right

auto

height

%

margin-top

auto

length

%

padding

padding-top

padding-right

padding-bottom

padding-left

padding-bottom

length

%

padding-left

length

%

padding-right

length

%

padding-top

length

%

display

none | inline | block | inline-block | flex | inline-flex | grid | inline-grid | contents | list-item |run-in | compact | table | inline-table | table-row-group | table-header-group | table-footer-group | table-row | table-column-group | table-column | table-cell | table-caption | ruby | ruby-base | ruby-text | ruby-base-group | ruby-text-group

overflow

visible | hidden | scroll |

auto | no-display | no-content

overflow-x

overflow-y

overflow-style

auto | marquee-line | marqueeblock

overflow-x

visible | hidden | scroll |

auto | no-display | no-content

visibility

visible | hidden | collapse

clear

left | right | both | none

Font

font

font-style

font-variant

font-weight

font-size/line-height

font-family

caption | icon | menu | messagebox | small-caption | status-bar

font-size-adjust

none | inherit

number

font-family

serif | sans-serif | Font Name

font-style

normal | italic | oblique | inherit

font-variant

normal | small-caps | inherit

font-size

xx-small | x-small | small | medium | large | x-large | xxlarge | smaller | larger |

inherit

length

%

font-weight

normal | bold | bolder | lighter | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 | inherit

Text

direction

ltr | rtl | inherit

hanging-punctuation

none | [start | end | endedge]

letter-spacing

normal

length

%

text-outline

none

color

length

unicode-bidi

normal | embed | bidi-override

white-space

normal | pre | nowrap | pre-wrap | pre-line

white-space-collapse

perserve | collapse | pre-servebreaks | discard

punctuation-trim

none | [start | end | adjacent]

text-align

start | end | left | right | center | justify

text-align-last

start | end | left | right | center | justify

text-decoration

none | underline | overline | line-thorugh | blink

text-shadow

none

color

length

word-break

normal | keep-all | loose | break-strict | break-all

word-wrap

normal

nowrap

text-emphasis

none | [[accent | dot | circle | disc | [before | after]?]

text-indent

length

%

text-justify

auto | inter-word | interideograph | inter-cluster | distribute | kashida | tibetan

text-transform

none | capitalize | uppercase | lowercase

text-wrap

normal | unresrricted | none | suppress

word-spacing

normal

length

%

Column

column-count

auto

number

column-fill

auto | balance

column-gap

normal

length

column-rule

column-rule-width

column-rule-style

column-rule-color

column-rule-style

```
border-style
```

columns

```
column-width

column-count
```

column-rule-width

```
thin | medium | thick

length
```

column-span

```
1 | all
```

column-width

```
auto

length
```

Colors

color

```
inherit

color
```

opacity

```
inherit

number
```

Table

border-collapse

```
collapse | separate
```

empty-cells

```
show | hide
```

border-spacing

```
length length
```

table-layout

```
auto | fixed
```

caption-side

```
top | bottom | left | right
```

Speech

cue

```
cue-before
```

```
cue-after
```

cue-before

```
url [ silent | x-soft | soft | medium | loud | x-loud | none | inherit
]
```

```
number
```

```
%
```

mark

```
mark-before
```

```
mark-after
```

mark-before

```
string
```

mark-after

```
string
```

voice-pitch-range

```
x-low | low | medium | high | xhigh | inherit
```

```
number
```

voice-stress

```
strong | moderate | none | reduced | inherit
```

voice-volume

```
silent | x-soft | soft | medium | loud | x-loud | inherit
```

```
number
```

```
%
```

cue-after

```
url [ silent | x-soft | soft | medium | loud | x-loud | none | inherit
]

number

%
```

pause

```
pause-before

pause-after
```

pause-before

```
none | x-weak | weak | medium | strong | x-strong | inherit

time
```

phonemes

```
string
```

voice-duration

```
time
```

voice-family

```
inherit | [ <specific-voice, age, generic-voice, number> ]
```

voice-rate

```
x-slow | slow | medium | fast | x-fast | inherit

%
```

voice-pitch

```
x-low | low | medium | high | xhigh | inherit

number

%
```

caption-side

```
top | bottom | left | right
```

rest

```
rest-before

rest-after
```

rest-before

none | x-weak | weak | medium | strong | x-strong | inherit

time

rest-after

none | x-weak | weak | medium | strong | x-strong | inherit

time

speak

none | normal | spell-out | digits | literal-punctuation | no-punctuation | inherit-number

List & Markers

list-style

list-style-type

list-style-position

list-style-image

list-style-image

none

url

list-style-type

none | asterisks | box | check | circle | diamond | disc | hyphen | square | decimal | decimalleading-zero | lower-roman | upper-roman | lower-alpha | upper-alpha | lower-greek | lower-latin | upper-latin | hebrew | armenian | georgian | cjk-ideographic | hiragana | katakana | hiragana-iroha | katakana-iroha | footnotes

marker-offset

auto

length

Animations

animations

animation-name

animation-duration

animation-timing-function

animation-delay

animation-iteration-count

animation-direction

animation-name

none | IDENT

animation-duration

time

animation-timing-function

ease | linear | ease-in | easeout | ease-in-out | cubic-Bezier (number, number, number, number)

animation-delay

time

animation-iteration-count

inherit

number

animation-direction

normal | alternate

animation-play-state

running | paused

rotation

angle

rotation-point

position (paired value off-set)

Transitions

transitions

```
transitions-property

transitions-duration

transitions-timing-function

transitions-delay
```

transitions-delay

```
time
```

transitions-duration

```
time
```

transitions-property

```
none | all
```

transition-timing-function

```
ease | linear | ease-in | ease-out | ease-in-out | cubicBezier(
number, number, number, number)
```

UI

appearance

```
normal | inherit | [icon | window | desktop | work-space | document |
tooltip | dialog | button | push-button | hyperlink | radio-button |
checkbox | menu-item | tab | menu | menubar | pull-down-menu |
pop-up-menu | list-menu | radio-group | checkbox-group | outline-tree
| range | field | combo-box | signature | password]
```

cursor

```
auto | crosshair | default | pointer | move | e-resize | neresize |
nw-resize | n-resize | se-resize | sw-resize | swresize | s-resize |
w-resize | text | wait | help

url
```

icon

```
auto | inherit

url
```

nav-index

```
auto | inherit

number
```

nav-up

```
auto | inherit <id> [current | root | <target-name>
```

nav-right

```
auto | inherit <id> [current | root | <target-name>
```

nav-down

```
auto | inherit <id> [current | root | <target-name>
```

nav-left

```
auto | inherit <id> [current |

root | <target-name>
```

resize

```
none | both | horizontal |

vertical | inherit
```

Pseudo-Class

`:active`	an activated element
`:focus`	an element while the element has focus
`:hover`	an element when you mouse over it
`:link`	an unvisited link
`:disabled`	an element while the element is disabled
`:enabled`	an element while the element is enabled
`:checked`	an element that is checked
`:selection`	an element that is currently selected or highlighted by the user
`:lang`	allows the author to specify a language to use in a specified element
`:nth-child(n)`	an element that is the n-th sibling
`:nth-last-child(n)`	an element that is the n-th sibling counting from the last sibling
`:first-child`	an element that is the first sibling
`:last-child`	an element that is the last sibling
`:only-child`	an element that is the only child
`:nth-of-type(n)`	an element that is the n-th sibling of its type
`:nth-last-of-type(n)`	an element that is the n-th sibling of its type counting from the last sibling
`:last-of-type`	an element that is the last sibling of its type
`:first-of-type`	an element that is the first sibling of its type
`:only-of-type`	an element that is the only child of its type
`:empty`	an element that has no children
`:root`	root element within the document
`:not(x)`	an element not represented by the argument ‘x’
`:target`	a target element as specified by a target in a UR

Pseudo-Element

`::first-letter`	Adds special style to the first letter of a text
`::first-line`	Adds special style to the first line of a text
`::before`	Inserts some content before the content of an element
`::after`	Inserts some content after the content of an element

Absolute Measurement

`%`	percentage
`cm`	centimeter
`in`	inch
`mm`	milimeter
`pc`	pica (1p = 12 points)
`pt`	point (1pt = 1/72 inch)
`px`	pixel (1px = 1/96 inch)

Relative Measurement

`ch`	width of the “0” glyph found in the font for the font size used to render
`em`	1em = current font size of current element
`ex`	x-height of the element’s font
`gd`	the grid defined by ‘layout-grid’
`rem`	the font size of the root element

`vh`	the viewport’s height
`vw`	the viewport's width
`vm`	viewport’s height or width, whichever is smaller of the two

Angles

`deg`	degrees
`grad`	grads
`rad`	radians
`turn`	turns

Time

`ms`	mili-seconds
`s`	seconds

Frequency

`Hz`	hertz
`kHz`	kilo-hertz

Colors

`color name`	red, blue, green, dark green
`rgb(x,y,z)`	red = rgb(255,0,0)
`rgb(x%,y%,z%)`	red = rgb(100%,0,0)
`rgba(x,y,z,alpha)`	red = rgba(255,0,0,0)
`#rrggbb`	red = #ff0000 (or shorthand - #f00)
`hsl(hue, saturation, lightness)`	red = hsl (0, 100%, 50%)
`flavor`	An accent color (typically chosen by the user) to customize the user interface of the user agent itself.
`currentColor`	computer value of the 'currentColor' keyword is the computed value of the 'color' property

Selector Types

Name	Info	Example
Universal	Any element	`* { font: 10px Arial; }`
Type	Any element of that type	`h1 { text-decoration: underline; }`
Grouping	Multiple elements of different types	`h1, h2, h3 { font-family: Verdana; }`
Class	Multiple elements of different types when you don't want to affect all instances of that type	`.sampleClass { text-decoration: underline; }`
Id	A single element type when you don't want to affect all instances of that type	`#sampleID { text-decoration: underline; }`
Descendant	An element that is below (in the document tree) another element - no matter how many levels	`#gallery h1 { text-decoration: underline; }`

	below	
Child	An element that is directly below (in the document tree) another element	`#title > p { font-weight: bold; }`
Adjacent Sibling	All elements that share the same parent and elements are in the same immediate sequence	`h1 + p { font-style: italic; }`
General Sibling	All elements that share the same parent and elements are in the same sequence (not necessarily immediate)	`h1 ~ p { font-style: italic; }`
Attribute	An element that matches the attribute listed	`E[selected] - att whatever the value E[att="val"] - att with a specific value E[rel~="next"] - att with a value is a whitespace separated list *[lang|="en"] - att value either being exactly "val" or beginning with "val" immediately followed by "-" E[att^="val"] - att value that begins with the prefix "val"`

Outline

outline

outline-color

outline-style

outline-width

outline-offset

inherit

length

outline-style

none | dotted | dashed | solid | double | groove | ridge | inset | outset

outline-width

thin | medium | thick

length

3D / 2D Transform

backface-visibility

visible | hidden

perspective

none

number

perspective-origin

[[percentage> | <length> | left | center | right] [<percentage> | <length> | top | center | bottom]?] <length>] | [[[left | center | right] || [top | center | bottom]] <length>]

transform

none | matrix | matrix3d | translate3d | tranlateX | translateY | translateZ | scale | scale3d | scaleX | scaleY | scaleZ | rotate | rotate3d | rotateX | rotateY | rotateZ | skewX | skewY | skew | perspective

transform-origin

```
[ [ [ <percentage> | <length> | left | center | right ] [ <percentage> | <length> | top | center | bottom ]? ] <length> ] | [ [ [ left | center | right ] || [ top | center | bottom ] ] <length> ]
```

transform-style

```
flat | preserve-3d
```

Generated Content

bookmark-label

```
content

attr

string
```

bookmark-level

```
none

integer
```

bookmark-target

```
self

url

attr
```

border-length

```
self

url

attr
```

content

```
normal | none | inhibit

url
```

counter-reset

```
none

identifier number
```

crop

auto

shape

display

normal | none | list-item

float-offset

length length

hyphenate-after

auto

integer

counter-increment

none

identifier number

hyphenate-lines

no-limit

integer

hyphenate-resource

none

url

hyphens

none | manual | auto

image-resolution

normal | auto

dpi

hyphenate-before

auto

integer

hyphenate-character

auto

string

marks

```
[crop || cross ] | none
```

move-to

```
normal | here

identifier
```

page-policy

```
start | first | last
```

quotes

```
none

string string string string
```

string-set

```
identifier

content-list
```

text-replace

```
none

[<string> <string>]+
```

Line Box

alignment-adjust

```
auto | baseline | before-edge | text-before-edge | middle | central |
after-edge | textafter-edge | ideographic | alphabetic | hanging |
mathematical

length

%
```

alignment-baseline

```
baseline | ise-script | beforeedge | text-before-edge | afteredge |
text-after-edge | central | middle | ideographic | alphabetic |
hanging | mathematical
```

baseline-shift

```
baseline | sub | super
```

length

%

dominant-baseline

auto | use-script | no-change | reset-size | alphabetic | hanging | ideographic | mathematical | central | middle | text-after-edge | text-beforeedge

drop-initial-after-align

alignment-baseline

drop-initial-after-align

central | middle | after-edge | text-after-edge | ideographic | alphabetic | mathematical

%

drop-initial-before-align

caps-height

alignment-baseline

drop-initial-before-adjust

before-edge | text-before-edge | central | middle | hanging | mathematical

length

%

drop-initial-value

initial

integer

drop-initial-size

auto

integer

%

line

inline-box-align

initial | last

integer

line-height

normal

number

length

%

line-stacking

line-stacking-strategy

line-stacking-ruby

line-stacking-shift

line-stacking-strategy

inline-line-height | block-lineheight | max-height | gridheight

line-stacking-ruby

exclude-ruby | include-ruby

line-stacking-shift

consider-shifts | disregardshifts

line-stacking

line-stacking-strategy

line-stacking-ruby

line-stacking-shift

text-height

auto | font-size | text-size |

max-size

vertical-align

Baseline | sub | super | top | text-top | middle | bottom | text-bottom

length

%

Hyperlink

target
target-name

target-new

target-position

target-name
current | root | parent | new |

modal

string

target-new
window | tab | none

target-position
above | behind | front | back

Positioning

bottom
auto

%

length

right
auto

%

length

clip
shape

auto

top
auto

```
%

length
```

left

```
auto

%

length
```

z-index

```
auto

Number
```

position

```
static | relative | absolute | fixed
```

Ruby

ruby-align

```
auto | start | left | center | end | right | distribute-letter |
distribute-space | line-edge
```

ruby-overhang

```
auto | start | end | none
```

ruby-position

```
before | after | right | inline
```

ruby-span

```
attr(x) | none
```

Paged Media

fit

```
fill | hidden | meet | slice
```

fit-position

```
[top | center | bottom] || [left | center | right]

length
```

```
%
```

orphans

```
integer
```

image-orientation

```
auto

angle
```

page

```
auto

identifier
```

page-break-after

```
auto | always | avoid | left | right
```

page-break-before

```
auto | always | avoid | left | right
```

page-break-inside

```
auto | avoid
```

size

```
auto | landscape | portrait

length
```

windows

```
integer
```

Beginner's Essential

Javascript Cheat Sheet

The language of the web.

WebsiteSetup

Table of Contents

Javascript Basics	2
Variables	2
Arrays	3
Operators	4
Functions	5
Loops	7
If - Else Statements	7
Strings	7
Regular Expressions	9
Numbers and Math	10
Dealing with Dates	12
DOM Node	14
Working with the Browser	18
Events	21
Errors	27

Javascript Basics

Including JavaScript in an HTML Page

```
<script type="text/javascript">
  //JS code goes here
</script>
```

Call an External JavaScript File

```
<script src="myscript.js"></script><code></code>
```

Including Comments

```
//
```

Single line comments

```
/* comment here */
```

Multi-line comments

Variables

var, const, let

```
var
```

The most common variable. Can be reassigned but only accessed within a function. Variables defined with var move to the top when code is executed.

```
const
```

Cannot be reassigned and not accessible before they appear within the code.

```
let
```

Similar to const, however, let variable can be reassigned but not re-declared.

Data Types

```
var age = 23
```

Numbers

```
var x
```

Variables

```
var a = "init"
```

Text (strings)

```
var b = 1 + 2 + 3
```

Operations

```
var c = true
```

True or false statements

```
const PI = 3.14
```

Constant numbers

```
var name = {firstName:"John", lastName:"Doe"}
```

Objects

Objects

```
var person = {
  firstName:"John",
  lastName:"Doe",
  age:20,
  nationality:"German"
};
```

Arrays

```
var fruit = ["Banana", "Apple", "Pear"];
```

Array Methods

`concat()`

Join several arrays into one

`indexOf()`

Returns the first position at which a given element appears in an array

`join()`

Combine elements of an array into a single string and return the string

`lastIndexOf()`

Gives the last position at which a given element appears in an array

`pop()`

Removes the last element of an array

`push()`

Add a new element at the end

`reverse()`

Reverse the order of the elements in an array

`shift()`

Remove the first element of an array

`slice()`

Pulls a copy of a portion of an array into a new array of 4 24

`sort()`

Sorts elements alphabetically

`splice()`

Adds elements in a specified way and position

`toString()`

Converts elements to strings

`unshift()`

Adds a new element to the beginning

`valueOf()`

Returns the primitive value of the specified object

Operators

Basic Operators

```
+     Addition
-     Subtraction
*     Multiplication
/     Division
(..)  Grouping operator
%     Modulus (remainder)
++    Increment numbers
--    Decrement numbers
```

Comparison Operators

```
==    Equal to
===   Equal value and equal type
!=    Not equal
!==   Not equal value or not equal type
>     Greater than
<     Less than
>=    Greater than or equal to
<=    Less than or equal to
?     Ternary operator
```

Logical Operators

```
&&    Logical and
||    Logical or
!     Logical not
```

Bitwise Operators

```
&     AND statement
|     OR statement
~     NOT
^     XOR
<<    Left shift
>>    Right shift
>>>   Zero fill right shift
```

Functions

```
function name(parameter1, parameter2, parameter3) {
  // what the function does
}
```

Outputting Data

`alert()`

Output data in an alert box in the browser window

`confirm()`

Opens up a yes/no dialog and returns true/false depending on user click

`console.log()`

Writes information to the browser console, good for debugging purposes

`document.write()`

Write directly to the HTML document

`prompt()`

Creates an dialogue for user input

Global Functions

`decodeURI()`

Decodes a Uniform Resource Identifier (URI) created by encodeURI or similar

`decodeURIComponent()`

Decodes a URI component

`encodeURI()`

Encodes a URI into UTF-8

`encodeURIComponent()`

Same but for URI components

`eval()`

Evaluates JavaScript code represented as a string

`isFinite()`

Determines whether a passed value is a finite number

`isNaN()`

Determines whether a value is NaN or not

`Number()`

Returns a number converted from its argument

`parseFloat()`

Parses an argument and returns a floating point number

`parseInt()`

Parses its argument and returns an integer

Loops

```
for (before loop; condition for loop; execute after loop) {
  // what to do during the loop
}
```

for

The most common way to create a loop in Javascript

while

Sets up conditions under which a loop executes

do while

Similar to the while loop, however, it executes at least once and performs a check at the end to see if the condition is met to execute again

break

Used to stop and exit the cycle at certain conditions

continue

Skip parts of the cycle if certain conditions are met of 7 24

If - Else Statements

```
if (condition) {
  // what to do if condition is met
} else {
  // what to do if condition is not met
}
```

Strings

```
var person = "John Doe";
```

Escape Characters

```
\'    – Single quote
\"    – Double quote
\\    – Backslash
\b    – Backspace
\f    – Form feed
\n    – New line
\r    – Carriage return
\t    – Horizontal tabulator
```

`\v` — Vertical tabulator

String Methods

`charAt()`

Returns a character at a specified position inside a string

`charCodeAt()`

Gives you the unicode of character at that position

`concat()`

Concatenates (joins) two or more strings into one

`fromCharCode()`

Returns a string created from the specified sequence of UTF-16 code units

`indexOf()`

Provides the position of the first occurrence of a specified text within a string

`lastIndexOf()`

Same as indexOf() but with the last occurrence, searching backwards

`match()`

Retrieves the matches of a string against a search pattern

`replace()`

Find and replace specific text in a string

`search()`

Executes a search for a matching text and returns its position

`slice()`

Extracts a section of a string and returns it as a new string

`split()`

Splits a string object into an array of strings at a specified position

`substr()`

Similar to slice() but extracts a substring depended on a specified number of characters

`substring()`

Also similar to slice() but can't accept negative indices

`toLowerCase()`

Convert strings to lowercase

`toUpperCase()`

Convert strings to uppercase

`valueOf()`

Returns the primitive value (that has no properties or methods) of a string object

Regular Expressions

Pattern Modifiers

```
e – Evaluate replacement
i – Perform case-insensitive matching
g – Perform global matching
m – Perform multiple line matching
s – Treat strings as single line
x – Allow comments and whitespace in pattern
U – Non Greedy pattern
```

Brackets

```
[abc]      Find any of the characters between the brackets
[^abc]     Find any character not in the brackets
[0-9]      Used to find any digit from 0 to 9
[A-z]      Find any character from uppercase A to lowercase z
(a|b|c)    Find any of the alternatives separated with |
```

Metacharacters

```
.    – Find a single character, except newline or line terminator
\w   – Word character
\W   – Non-word character
\d   – A digit
\D   – A non-digit character
\s   – Whitespace character
\S   – Non-whitespace character
\b   – Find a match at the beginning/end of a word
\B   – A match not at the beginning/end of a word
\0   – NUL character
\n   – A new line character
\f   – Form feed character
\r   – Carriage return character
\t   – Tab character
\v   – Vertical tab character
```

```
\xxx — The character specified by an octal number xxx
\xdd — Character specified by a hexadecimal number dd
\uxxxx — The Unicode character specified by a hexadecimal number xxxx
```

Quantifiers

```
n+    — Matches any string that contains at least one n
n*    — Any string that contains zero or more occurrences of n
n?    — A string that contains zero or one occurrences of n
n{X}  — String that contains a sequence of X n's
n{X,Y} — Strings that contains a sequence of X to Y n's
n{X,} — Matches any string that contains a sequence of at least X n's
n$    — Any string with n at the end of it
^n    — String with n at the beginning of it
?=n   — Any string that is followed by a specific string n
?!n   — String that is not followed by a specific string n
```

Numbers and Math

Number Properties

MAX_VALUE

The maximum numeric value representable in JavaScript

MIN_VALUE

Smallest positive numeric value representable in JavaScript

NaN

The "Not-a-Number" value

NEGATIVE_INFINITY

The negative Infinity value

POSITIVE_INFINITY

Positive Infinity value

Number Methods

toExponential()

Returns a string with a rounded number written as exponential notation

toFixed()

Returns the string of a number with a specified number of decimals

`toPrecision()`

String of a number written with a specified length

`toString()`

Returns a number as a string

`valueOf()`

Returns a number as a number

Math Properties

```
E          Euler's number
LN2        The natural logarithm of 2
LN10       Natural logarithm of 10
LOG2E      Base 2 logarithm of E
LOG10E     Base 10 logarithm of E
PI         The number PI
SQRT1_2    Square root of 1/2
SQRT2      The square root of 2
```

Math Methods

`abs(x)`

Returns the absolute (positive) value of x

`acos(x)`

The arccosine of x, in radians

`asin(x)`

Arcsine of x, in radians

`atan(x)`

The arctangent of x as a numeric value

`atan2(y,x)`

Arctangent of the quotient of its arguments

`ceil(x)`

Value of x rounded up to its nearest integer

`cos(x)`

The cosine of x (x is in radians)

`exp(x)`

Value of Ex

`floor(x)`

The value of x rounded down to its nearest integer

`log(x)`

The natural logarithm (base E) of x

`max(x,y,z,...,n)`

Returns the number with the highest value

`min(x,y,z,...,n)`

Same for the number with the lowest value

`pow(x,y)`

X to the power of y

`random()`

Returns a random number between 0 and 1

`round(x)`

The value of x rounded to its nearest integer

`sin(x)`

The sine of x (x is in radians)

`sqrt(x)`

Square root of x

`tan(x)`

The tangent of an angle

Dealing with Dates

Setting Dates

`Date()`

Creates a new date object with the current date and time

```
Date(2017, 5, 21, 3, 23, 10, 0)
```

Create a custom date object. The numbers represent year, month, day, hour, minutes, seconds, milliseconds. You can omit anything you want except for year and month.

```
Date("2017-06-23")
```

Date declaration as a string

Pulling Date and Time Values

`getDate()`

Get the day of the month as a number (1-31)

`getDay()`

The weekday as a number (0-6)

`getFullYear()`

Year as a four digit number (yyyy)

`getHours()`

Get the hour (0-23)

`getMilliseconds()`

The millisecond (0-999)

`getMinutes()`

Get the minute (0-59)

`getMonth()`

Month as a number (0-11)

`getSeconds()`

Get the second (0-59)

`getTime()`

Get the milliseconds since January 1, 1970

`getUTCDate()`

The day (date) of the month in the specified date according to universal time (also available for day, month, fullyear, hours, minutes etc.)

`parse`

Parses a string representation of a date, and returns the number of milliseconds since January 1, 1970

Set Part of a Date

`setDate()`

Set the day as a number (1-31)

`setFullYear()`

Sets the year (optionally month and day)

`setHours()`

Set the hour (0-23)

`setMilliseconds()`

Set milliseconds (0-999)

`setMinutes()`

Sets the minutes (0-59)

`setMonth()`

Set the month (0-11)

`setSeconds()`

Sets the seconds (0-59)

`setTime()`

Set the time (milliseconds since January 1, 1970)

`setUTCDate()`

Sets the day of the month for a specified date according to universal time (also available for day, month, fullyear, hours, minutes etc.)

DOM Node

Node Properties

`attributes`

Returns a live collection of all attributes registered to and element

`baseURI`

Provides the absolute base URL of an HTML element

`childNodes`

Gives a collection of an element's child nodes

`firstChild`

Returns the first child node of an element

`lastChild`

The last child node of an element

`nextSibling`

Gives you the next node at the same node tree level

`nodeName`

Returns the name of a node

`nodeType`

Returns the type of a node

`nodeValue`

Sets or returns the value of a node

`ownerDocument`

The top-level document object for this node

`parentNode`

Returns the parent node of an element

`previousSibling`

Returns the node immediately preceding the current one

`textContent`

Sets or returns the textual content of a node and its descendants

Node Methods

`appendChild()`

Adds a new child node to an element as the last child node

`cloneNode()`

Clones an HTML element

`compareDocumentPosition()`

Compares the document position of two elements

`getFeature()`

Returns an object which implements the APIs of a specified feature

`hasAttributes()`

Returns true if an element has any attributes, otherwise false

`hasChildNodes()`

Returns true if an element has any child nodes, otherwise false

`insertBefore()`

Inserts a new child node before a specified, existing child node

`isDefaultNamespace()`

Returns true if a specified namespaceURI is the default, otherwise false

`isEqualNode()`

Checks if two elements are equal

`isSameNode()`

Checks if two elements are the same node

`isSupported()`

Returns true if a specified feature is supported on the element

`lookupNamespaceURI()`

Returns the namespaceURI associated with a given node

`lookupPrefix()`

Returns a DOMString containing the prefix for a given namespaceURI, if present

`normalize()`

Joins adjacent text nodes and removes empty text nodes in an element

`removeChild()`

Removes a child node from an element

`replaceChild()`

Replaces a child node in an element

Element Methods

`getAttribute()`

Returns the specified attribute value of an element node

`getAttributeNS()`

Returns string value of the attribute with the specified namespace and name

`getAttributeNode()`

Gets the specified attribute node

`getAttributeNodeNS()`

Returns the attribute node for the attribute with the given namespace and name

`getElementsByTagName()`

Provides a collection of all child elements with the specified tag name

`getElementsByTagNameNS()`

Returns a live HTMLCollection of elements with a certain tag name belonging to the given namespace

`hasAttribute()`

Returns true if an element has any attributes, otherwise false

`hasAttributeNS()`

Provides a true/false value indicating whether the current element in a given namespace has the specified attribute

`removeAttribute()`

Removes a specified attribute from an element

`removeAttributeNS()`

Removes the specified attribute from an element within a certain namespace

`removeAttributeNode()`

Takes away a specified attribute node and returns the removed node

`setAttribute()`

Sets or changes the specified attribute to a specified value

`setAttributeNS()`

Adds a new attribute or changes the value of an attribute with the given namespace and name

`setAttributeNode()`

Sets or changes the specified attribute node

`setAttributeNodeNS()`

Adds a new namespaced attribute node to an element

Working with the Browser

Window Properties

closed

Checks whether a window has been closed or not and returns true or false

defaultStatus

Sets or returns the default text in the statusbar of a window

document

Returns the document object for the window

frames

Returns all <iframe> elements in the current window

history

Provides the History object for the window

innerHeight

The inner height of a window's content area

innerWidth

The inner width of the content area

length

Find out the number of <iframe> elements in the window

location

Returns the location object for the window

name

Sets or returns the name of a window

navigator

Returns the Navigator object for the window

opener

Returns a reference to the window that created the window

outerHeight

The outer height of a window, including toolbars/ scrollbars

`outerWidth`

The outer width of a window, including toolbars/ scrollbars

`pageXOffset`

Number of pixels the current document has been scrolled horizontally

`pageYOffset`

Number of pixels the document has been scrolled vertically

`parent`

The parent window of the current window

`screen`

Returns the Screen object for the window

`screenLeft`

The horizontal coordinate of the window (relative to screen)

`screenTop`

The vertical coordinate of the window

`screenX`

Same as screenLeft but needed for some browsers

`screenY`

Same as screenTop but needed for some browsers

`self`

Returns the current window

`status`

Sets or returns the text in the statusbar of a window

`top`

Returns the topmost browser window

Window Methods

`alert()`

Displays an alert box with a message and an OK button

`blur()`

Removes focus from the current window

`clearInterval()`

Clears a timer set with setInterval()

`clearTimeout()`

Clears a timer set with setTimeout()

`close()`

Closes the current window

`confirm()`

Displays a dialogue box with a message and an OK and Cancel button

`focus()`

Sets focus to the current window

`moveBy()`

Moves a window relative to its current position

`moveTo()`

Moves a window to a specified position

`open()`

Opens a new browser window

`print()`

Prints the content of the current window

`prompt()`

Displays a dialogue box that prompts the visitor for input

`resizeBy()`

Resizes the window by the specified number of pixels

`resizeTo()`

Resizes the window to a specified width and height

`scrollBy()`

Scrolls the document by a specified number of pixels

`scrollTo()`

Scrolls the document to specific coordinates

`setInterval()`

Calls a function or evaluates an expression at specified intervals

`setTimeout()`

Calls a function or evaluates an expression after a specified interval

`stop()`

Stops the window from loading

Screen Properties

`availHeight`

Returns the height of the screen (excluding the Windows Taskbar)

`availWidth`

Returns the width of the screen (excluding the Windows Taskbar)

`colorDepth`

Returns the bit depth of the color palette for displaying images

`height`

The total height of the screen

`pixelDepth`

The color resolution of the screen in bits per pixel

`width`

The total width of the screen

Events

Mouse

`onclick`

The event occurs when the user clicks on an element

`oncontextmenu`

User right-clicks on an element to open a context menu

`ondblclick`

The user double-clicks on an element

onmousedown

User presses a mouse button over an element

onmouseenter

The pointer moves onto an element

onmouseleave

Pointer moves out of an element

onmousemove

The pointer is moving while it is over an element

onmouseover

When the pointer is moved onto an element or one of its children

onmouseout

User moves the mouse pointer out of an element or one of its children

onmouseup

The user releases a mouse button while over an element

Keyboard

onkeydown

When the user is pressing a key down

onkeypress

The moment the user starts pressing a key

onkeyup

The user releases a key

Frame

onabort

The loading of a media is aborted

onbeforeunload

Event occurs before the document is about to be unloaded

onerror

An error occurs while loading an external file

onhashchange

There have been changes to the anchor part of a URL

onload

When an object has loaded

onpagehide

The user navigates away from a webpage

onpageshow

When the user navigates to a webpage

onresize

The document view is resized

onscroll

An element's scrollbar is being scrolled

onunload

Event occurs when a page has unloaded

Form

onblur

When an element loses focus

onchange

The content of a form element changes (for <input>, <select>and <textarea>)

onfocus

An element gets focus

onfocusin

When an element is about to get focus

onfocusout

The element is about to lose focus

oninput

User input on an element

oninvalid

An element is invalid

onreset

A form is reset

onsearch

The user writes something in a search field (for <input="search">)

onselect

The user selects some text (for <input> and <textarea>)

onsubmit

A form is submitted

Drag

ondrag

An element is dragged

ondragend

The user has finished dragging the element

ondragenter

The dragged element enters a drop target

ondragleave

A dragged element leaves the drop target

ondragover

The dragged element is on top of the drop target

ondragstart

User starts to drag an element

ondrop

Dragged element is dropped on the drop target

Clipboard

oncopy

User copies the content of an element

oncut

The user cuts an element's content

`onpaste`

A user pastes content in an element

Media

`onabort`

Media loading is aborted

`oncanplay`

The browser can start playing media (e.g. a file has buffered enough)

`oncanplaythrough`

When browser can play through media without stopping

`ondurationchange`

The duration of the media changes

`onended`

The media has reached its end

`onerror`

Happens when an error occurs while loading an external file

`onloadeddata`

Media data is loaded

`onloadedmetadata`

Meta Metadata (like dimensions and duration) are loaded

`onloadstart`

Browser starts looking for specified media

`onpause`

Media is paused either by the user or automatically

`onplay`

The media has been started or is no longer paused

`onplaying`

Media is playing after having been paused or stopped for buffering

`onprogress`

Browser is in the process of downloading the media

onratechange

The playing speed of the media changes

onseeked

User is finished moving/skipping to a new position in the media

onseeking

The user starts moving/skipping

onstalled

The browser is trying to load the media but it is not available

onsuspend

Browser is intentionally not loading media

ontimeupdate

The playing position has changed (e.g. because of fast forward)

onvolumechange

Media volume has changed (including mute)

onwaiting

Media paused but expected to resume (for example, buffering)

Animation

animationend

A CSS animation is complete

animationiteration

CSS animation is repeated

animationstart

CSS animation has started

Other

transitionend

Fired when a CSS transition has completed

onmessage

A message is received through the event source

onoffline

Browser starts to work offline

ononline

The browser starts to work online

onpopstate

When the window's history changes

onshow

A <menu> element is shown as a context menu

onstorage

A Web Storage area is updated

ontoggle

The user opens or closes the <details> element

onwheel

Mouse wheel rolls up or down over an element

ontouchcancel

Screen touch is interrupted

ontouchend

User finger is removed from a touch screen

ontouchmove

A finger is dragged across the screen

ontouchstart

Finger is placed on touch screen

Errors

try

Lets you define a block of code to test for errors

catch

Set up a block of code to execute in case of an error

throw

Create custom error messages instead of the standard JavaScript errors

finally

Lets you execute code, after try and catch, regardless of the result

Error Name Values

name

Sets or returns the error name

message

Sets or returns an error message in string from

EvalError

An error has occurred in the eval() function

RangeError

A number is “out of range”

ReferenceError

An illegal reference has occurred

SyntaxError

A syntax error has occurred

TypeError

A type error has occurred

URIError

An encodeURI() error has occurred