Document

- Constructor
- Instance properties
  - Extensions for HTMLDocument
  - Deprecated properties
- Instance methods
  - Extension for HTML documents
- Static methods
- Events
  - Fullscreen events
  - Load & unload events
  - Pointer lock events
  - Scroll events
  - Selection events
  - Bubbled events
- Specifications
- Browser compatibility

{{APIRef("DOM")}}

The **`Document`** interface represents any web page loaded in the browser and serves as an entry point into the web page's content, which is the DOM tree.

The DOM tree includes elements such as {{HTMLElement("body")}} and {{HTMLElement("table")}}, among many others. It provides functionality globally to the document, like how to obtain the page's URL and create new elements in the document.

{{InheritanceDiagram}}

The `Document` interface describes the common properties and methods for any kind of document. Depending on the document's type (e.g., HTML, XML, SVG, …), a larger API is available: HTML documents, served with the `"text/html"` content type, also implement the

{{DOMxRef("HTMLDocument")}} interface, whereas XML and SVG documents implement the {{DOMxRef("XMLDocument")}} interface.

Constructor

- {{DOMxRef("Document.Document", "Document()")}}
  - : Creates a new `Document` object.

Instance properties

This interface also inherits from the {{DOMxRef("Node")}} and {{DOMxRef("EventTarget")}} interfaces.

- {{DOMxRef("Document.activeElement")}} {{ReadOnlyInline}}
  - : Returns the {{DOMxRef('Element')}} that currently has focus.
- {{DOMxRef("Document.activeViewTransition")}} {{ReadOnlyInline}}
  - : Returns a {{DOMxRef('ViewTransition')}} instance representing the view transition currently active on the document, or `null` if there is no active view transition.
- {{DOMxRef("Document.adoptedStyleSheets")}}
  - : Add an array of constructed stylesheets to be used by the document. These stylesheets may also be shared with shadow DOM subtrees of the same document.
- {{DOMxRef("Document.body")}}
  - : Returns the {{HTMLElement("body")}} or {{htmlelement("frameset")}} node of the current document.
- {{DOMxRef("Document.characterSet")}} {{ReadOnlyInline}}
  - : Returns the character set being used by the document.
- {{domxref("Document.childElementCount")}} {{ReadOnlyInline}}
  - : Returns the number of child elements of the current document.
- {{domxref("Document.children")}} {{ReadOnlyInline}}
  - : Returns the child elements of the current document.
- {{DOMxRef("Document.compatMode")}} {{ReadOnlyInline}}
  - : Indicates whether the document is rendered in *quirks* or *strict* mode.
- {{DOMxRef("Document.contentType")}} {{ReadOnlyInline}}
  - : Returns the Content-Type from the MIME Header of the current document.
- {{DOMxRef("Document.currentScript")}} {{ReadOnlyInline}}
  - : Returns the {{HTMLElement("script")}} element whose script is currently being processed and isn't a JavaScript module.
- {{DOMxRef("Document.customElementRegistry")}} {{ReadOnlyInline}}
  - : The {{domxref("CustomElementRegistry")}} object associated with this document, or `null` if one has not been set.

- {{DOMxRef("Document.doctype")}} {{ReadOnlyInline}}
  - : Returns the Document Type Definition (DTD) of the current document.
- {{DOMxRef("Document.documentElement")}} {{ReadOnlyInline}}
  - : Returns the {{DOMxRef("Element")}} that is a direct child of the document. For HTML documents, this is normally the {{DOMxRef("HTMLHtmlElement")}} object representing the document's {{HTMLElement("html")}} element.
- {{DOMxRef("Document.documentURI")}} {{ReadOnlyInline}}
  - : Returns the document location as a string.
- {{DOMxRef("Document.embeds")}} {{ReadOnlyInline}}
  - : Returns an {{DOMxRef("HTMLCollection")}} of the embedded {{HTMLElement('embed')}} elements in the document.
- {{DOMxRef("Document.featurePolicy")}} {{Experimental_Inline}} {{ReadOnlyInline}} {{non-standard_inline}}
  - : Returns the {{DOMxRef("FeaturePolicy")}} interface with the feature policies applied to the document.
- {{domxref("Document.firstElementChild")}} {{ReadOnlyInline}}
  - : Returns the first child element of the current document.
- {{DOMxRef("Document.fonts")}} {{ReadOnlyInline}}
  - : Returns the {{DOMxRef("FontFaceSet")}} interface of the current document.
- {{DOMxRef("Document.forms")}} {{ReadOnlyInline}}
  - : Returns an {{DOMxRef("HTMLCollection")}} of the {{HTMLElement("form")}} elements in the document.
- {{DOMxRef("Document.fragmentDirective")}} {{ReadOnlyInline}}
  - : Returns the {{domxref("FragmentDirective")}} for the current document.
- {{DOMxRef("Document.fullscreenElement")}} {{ReadOnlyInline}}
  - : The element that's currently in fullscreen mode for this document.
- {{DOMxRef("Document.head")}} {{ReadOnlyInline}}
  - : Returns the {{HTMLElement("head")}} element of the current document.
- {{DOMxRef("Document.hidden")}} {{ReadOnlyInline}}
  - : Returns a Boolean value indicating if the page is considered hidden or not.
- {{DOMxRef("Document.images")}} {{ReadOnlyInline}}
  - : Returns an {{DOMxRef("HTMLCollection")}} of the images in the document.
- {{DOMxRef("Document.implementation")}} {{ReadOnlyInline}}
  - : Returns the DOM implementation associated with the current document.
- {{domxref("Document.lastElementChild")}} {{ReadOnlyInline}}
  - : Returns the last child element of the current document.
- {{DOMxRef("Document.links")}} {{ReadOnlyInline}}
  - : Returns an {{DOMxRef("HTMLCollection")}} of the hyperlinks in the document.
- {{DOMxRef("Document.pictureInPictureElement")}} {{ReadOnlyInline}}
  - : Returns the {{DOMxRef('Element')}} currently being presented in picture-in-picture mode in this document.
- {{DOMxRef("Document.pictureInPictureEnabled")}} {{ReadOnlyInline}}
  - : Returns true if the picture-in-picture feature is enabled.
- {{DOMxRef("Document.plugins")}} {{ReadOnlyInline}}
  - : Returns an {{DOMxRef("HTMLCollection")}} of the available plugins.

- {{DOMxRef("Document.pointerLockElement")}} {{ReadOnlyInline}}
  - : Returns the element set as the target for mouse events while the pointer is locked. `null` if lock is pending, pointer is unlocked, or if the target is in another document.
- {{DOMxRef("Document.prerendering")}} {{ReadOnlyInline}} {{experimental_inline}}
  - : Returns a boolean that indicates whether the document is currently in the process of prerendering, as initiated via the Speculation Rules API.
- {{DOMxRef("Document.scripts")}} {{ReadOnlyInline}}
  - : Returns an {{DOMxRef("HTMLCollection")}} of the {{HTMLElement("script")}} elements in the document.
- {{DOMxRef("Document.scrollingElement")}} {{ReadOnlyInline}}
  - : Returns a reference to the {{DOMxRef("Element")}} that scrolls the document.
- {{DOMxRef("Document.styleSheets")}} {{ReadOnlyInline}}
  - : Returns a {{DOMxRef('StyleSheetList')}} of {{DOMxRef('CSSStyleSheet')}} objects for stylesheets explicitly linked into, or embedded in a document.
- {{DOMxRef("Document.timeline")}} {{ReadOnlyInline}}
  - : Returns timeline as a special instance of {{domxref("DocumentTimeline")}} that is automatically created on page load.
- {{DOMxRef("Document.visibilityState")}} {{ReadOnlyInline}}
  - : Returns a `string` denoting the visibility state of the document. Possible values are `visible`, `hidden`, and `unloaded`.

Extensions for HTMLDocument

The `Document` interface for HTML documents inherits from the {{DOMxRef("HTMLDocument")}} interface or is extended for such documents.

- {{DOMxRef("Document.cookie")}}
  - : Returns a semicolon-separated list of the cookies for that document or sets a single cookie.
- {{DOMxRef("Document.defaultView")}} {{ReadOnlyInline}}
  - : Returns a reference to the window object.
- {{DOMxRef("Document.designMode")}}
  - : Gets/sets the ability to edit the whole document.
- {{DOMxRef("Document.dir")}}
  - : Gets/sets directionality (rtl/ltr) of the document.
- {{DOMxRef("Document.fullscreenEnabled")}} {{ReadOnlyInline}}
  - : Indicates whether fullscreen mode is available.
- {{DOMxRef("Document.lastModified")}} {{ReadOnlyInline}}
  - : Returns the date on which the document was last modified.
- {{DOMxRef("Document.location")}} {{ReadOnlyInline}}
  - : Returns the URI of the current document.
- {{DOMxRef("Document.readyState")}} {{ReadOnlyInline}}
  - : Returns loading status of the document.
- {{DOMxRef("Document.referrer")}} {{ReadOnlyInline}}

    - : Returns the URI of the page that linked to this page.
- {{DOMxRef("Document.title")}}
    - : Sets or gets the title of the current document.
- {{DOMxRef("Document.URL")}} {{ReadOnlyInline}}
    - : Returns the document location as a string.
- Named properties
    - : Some elements in the document are also exposed as properties:
        - For each {{HTMLElement("embed")}}, {{HTMLElement("form")}}, {{HTMLElement("iframe")}}, {{HTMLElement("img")}}, and {{HTMLElement("object")}} element, its `name` (if non-empty) is exposed. For example, if the document contains `<form name="my_form">`, then `document["my_form"]` (and its equivalent `document.my_form`) returns a reference to that element.
        - For each {{HTMLElement("object")}} element, its `id` (if non-empty) is exposed.
        - For each {{HTMLElement("img")}} element with non-empty `name`, its `id` (if non-empty) is exposed.

      If a property corresponds to a single element, that element is directly returned. If that single element is an iframe, then its {{domxref("HTMLIFrameElement/contentWindow", "contentWindow")}} is returned instead. If the property corresponds to multiple elements, then an {{domxref("HTMLCollection")}} is returned containing all of them.

Deprecated properties

- {{DOMxRef("Document.alinkColor")}} {{Deprecated_Inline}}
    - : Returns or sets the color of active links in the document body.
- {{DOMxRef("Document.all")}} {{ReadOnlyInline}} {{Deprecated_Inline}}
    - : Provides access to all elements in the document — it returns an {{DOMxRef('HTMLAllCollection')}} rooted at the document node. This is a legacy, non-standard property and should not be used.
- {{DOMxRef("Document.anchors")}} {{Deprecated_Inline}} {{ReadOnlyInline}}
    - : Returns a list of all of the anchors in the document.
- {{DOMxRef("Document.applets")}} {{Deprecated_Inline}} {{ReadOnlyInline}}
    - : Returns an empty {{domxref("HTMLCollection")}}. Legacy property that used to return the list of applets within a document.
- {{DOMxRef("Document.bgColor")}} {{Deprecated_Inline}}
    - : Gets/sets the background color of the current document.
- {{DOMxRef("Document.characterSet","Document.charset")}} {{Deprecated_Inline}} {{ReadOnlyInline}}
    - : Alias of {{DOMxRef("Document.characterSet")}}. Use this property instead.
- {{DOMxRef("Document.domain")}} {{Deprecated_Inline}}
    - : Gets/sets the domain of the current document.
- {{DOMxRef("Document.fgColor")}} {{Deprecated_Inline}}

    - : Gets/sets the foreground color, or text color, of the current document.
- {{DOMxRef("Document.fullscreen")}} {{ReadOnlyInline}} {{Deprecated_Inline}}
    - : Returns `true` when the document is in fullscreen mode.
- {{DOMxRef("Document.characterSet", "Document.inputEncoding")}} {{Deprecated_Inline}} {{ReadOnlyInline}}
    - : Alias of {{DOMxRef("Document.characterSet")}}. Use this property instead.
- {{DOMxRef("Document.lastStyleSheetSet")}} {{Deprecated_Inline}} {{ReadOnlyInline}} {{Non-standard_Inline}}
    - : Returns the name of the style sheet set that was last enabled. Has the value `null` until the style sheet is changed by setting the value of {{DOMxRef("Document.selectedStyleSheetSet","selectedStyleSheetSet")}}.
- {{DOMxRef("Document.linkColor")}} {{Deprecated_Inline}}
    - : Gets/sets the color of hyperlinks in the document.
- {{DOMxRef("Document.preferredStyleSheetSet")}} {{Deprecated_Inline}} {{ReadOnlyInline}} {{Non-standard_Inline}}
    - : Returns the preferred style sheet set as specified by the page author.
- {{DOMxRef("Document.rootElement")}} {{ReadOnlyInline}} {{Deprecated_Inline}}
    - : Like {{DOMxRef("Document.documentElement")}}, but only for {{SVGElement("svg")}} root elements. Use this property instead.
- {{DOMxRef("Document.selectedStyleSheetSet")}} {{Deprecated_Inline}} {{Non-standard_Inline}}
    - : Returns which style sheet set is currently in use.
- {{DOMxRef("Document.styleSheetSets")}} {{Deprecated_Inline}} {{ReadOnlyInline}} {{Non-standard_Inline}}
    - : Returns a list of the style sheet sets available on the document.
- {{DOMxRef("Document.vlinkColor")}} {{Deprecated_Inline}}
    - : Gets/sets the color of visited hyperlinks.
- {{DOMxRef("Document.xmlEncoding")}} {{ReadOnlyInline}} {{Deprecated_Inline}}
    - : Returns the encoding as determined by the XML declaration.
- `Document.xmlStandalone` {{Deprecated_Inline}}
    - : Returns `true` if the XML declaration specifies the document to be standalone (*e.g.,* An external part of the DTD affects the document's content), else `false`.
- {{DOMxRef("Document.xmlVersion")}} {{Deprecated_Inline}}
    - : Returns the version number as specified in the XML declaration or `"1.0"` if the declaration is absent.

Instance methods

This interface also inherits from the {{DOMxRef("Node")}} and {{DOMxRef("EventTarget")}} interfaces.

- {{DOMxRef("Document.adoptNode()")}}
    - : Adopt node from an external document.

- {{DOMxRef("Document.append()")}}
  - : Inserts a set of {{domxref("Node")}} objects or strings after the last child of the document.
- {{DOMxRef("Document.ariaNotify()")}}
  - : Specifies that a given string of text should be announced by a screen reader.
- {{DOMxRef("Document.browsingTopics()")}} {{non-standard_inline}} {{deprecated_inline}}
  - : Returns a promise that fulfills with an array of objects representing the top topics for the user, one from each of the last three epochs. By default, the method also causes the browser to record the current page visit as observed by the caller, so the page's hostname can later be used in topics calculation. See the Topics API for more details.
- `Document.captureEvents()` {{Deprecated_Inline}}
  - : See {{DOMxRef("Window.captureEvents")}}.
- {{DOMxRef("Document.caretPositionFromPoint()")}}
  - : Returns a {{DOMxRef('CaretPosition')}} object containing the DOM node containing the caret, and caret's character offset within that node.
- {{DOMxRef("Document.caretRangeFromPoint()")}} {{Non-standard_Inline}}
  - : Gets a {{DOMxRef("Range")}} object for the document fragment under the specified coordinates.
- {{DOMxRef("Document.createAttribute()")}}
  - : Creates a new {{DOMxRef("Attr")}} object and returns it.
- {{DOMxRef("Document.createAttributeNS()")}}
  - : Creates a new attribute node in a given namespace and returns it.
- {{DOMxRef("Document.createCDATASection()")}}
  - : Creates a new CDATA node and returns it.
- {{DOMxRef("Document.createComment()")}}
  - : Creates a new comment node and returns it.
- {{DOMxRef("Document.createDocumentFragment()")}}
  - : Creates a new document fragment.
- {{DOMxRef("Document.createElement()")}}
  - : Creates a new element with the given tag name.
- {{DOMxRef("Document.createElementNS()")}}
  - : Creates a new element with the given tag name and namespace URI.
- {{DOMxRef("Document.createEvent()")}} {{deprecated_inline}}
  - : Creates an event object.
- {{DOMxRef("Document.createNodeIterator()")}}
  - : Creates a {{DOMxRef("NodeIterator")}} object.
- {{DOMxRef("Document.createProcessingInstruction()")}}
  - : Creates a new {{DOMxRef("ProcessingInstruction")}} object.
- {{DOMxRef("Document.createRange()")}}
  - : Creates a {{DOMxRef("Range")}} object.
- {{DOMxRef("Document.createTextNode()")}}
  - : Creates a text node.

- {{DOMxRef("Document.createTouch()")}} {{Deprecated_Inline}} {{Non-standard_Inline}}
  - : Creates a {{DOMxRef("Touch")}} object.
- {{DOMxRef("Document.createTouchList()")}} {{Deprecated_Inline}} {{Non-standard_Inline}}
  - : Creates a {{DOMxRef("TouchList")}} object.
- {{DOMxRef("Document.createTreeWalker()")}}
  - : Creates a {{DOMxRef("TreeWalker")}} object.
- {{DOMxRef("Document.elementFromPoint()")}}
  - : Returns the topmost element at the specified coordinates.
- {{DOMxRef("Document.elementsFromPoint()")}}
  - : Returns an array of all elements at the specified coordinates.
- {{DOMxRef("Document.enableStyleSheetsForSet()")}} {{Deprecated_Inline}} {{Non-standard_Inline}}
  - : Enables the style sheets for the specified style sheet set.
- {{DOMxRef("Document.exitFullscreen()")}}
  - : Stops document's fullscreen element from being displayed fullscreen.
- {{DOMxRef("Document.exitPictureInPicture()")}}
  - : Remove the video from the floating picture-in-picture window back to its original container.
- {{DOMxRef("Document.exitPointerLock()")}}
  - : Release the pointer lock.
- {{DOMxRef("Document.getAnimations()")}}
  - : Returns an array of all {{DOMxRef("Animation")}} objects currently in effect, whose target elements are descendants of the `document`.
- `Document.getBoxQuads()` {{Experimental_Inline}}
  - : Returns a list of {{domxref("DOMQuad")}} objects representing the CSS fragments of the node.
- {{DOMxRef("Document.getElementById", "Document.getElementById()")}}
  - : Returns an object reference to the identified element.
- {{DOMxRef("Document.getElementsByClassName()")}}
  - : Returns a list of elements with the given class name.
- {{DOMxRef("Document.getElementsByTagName()")}}
  - : Returns a list of elements with the given tag name.
- {{DOMxRef("Document.getElementsByTagNameNS()")}}
  - : Returns a list of elements with the given tag name and namespace.
- {{DOMxRef("Document.getSelection()")}}
  - : Returns a {{DOMxRef('Selection')}} object representing the range of text selected by the user, or the current position of the caret.
- {{domxref("Document.hasPrivateToken()")}} {{experimental_inline}}
  - : Returns a promise that fulfills with a boolean indicating whether the browser has a private state token stored from a particular issuer.
- {{domxref("Document.hasRedemptionRecord()")}} {{experimental_inline}}
  - : Returns a promise that fulfills with a boolean indicating whether the browser has a redemption record originating from a particular issuer.

- {{DOMxRef("Document.hasStorageAccess()")}}
  - : Returns a {{jsxref("Promise")}} that resolves with a boolean value indicating whether the document has access to unpartitioned cookies.
- {{DOMxRef("Document.hasUnpartitionedCookieAccess()")}}
  - : New name for {{DOMxRef("Document.hasStorageAccess()")}}.
- {{DOMxRef("Document.importNode()")}}
  - : Returns a clone of a node from an external document.
- {{DOMxRef("Document.moveBefore()")}}
  - : Moves a given {{domxref("Node")}} inside the `Document` DOM node as a direct child, before a given reference node, without removing and then inserting the node.
- {{DOMxRef("Document.mozSetImageElement()")}} {{Non-standard_Inline}}
  - : Allows you to change the element being used as the background image for a specified element ID.
- {{DOMxRef("Document.prepend()")}}
  - : Inserts a set of {{domxref("Node")}} objects or strings before the first child of the document.
- {{DOMxRef("Document.querySelector()")}}
  - : Returns the first Element node within the document, in document order, that matches the specified selectors.
- {{DOMxRef("Document.querySelectorAll()")}}
  - : Returns a list of all the Element nodes within the document that match the specified selectors.
- {{DOMxRef("Document.releaseCapture()")}} {{Non-standard_Inline}}
  - : Releases the current mouse capture if it's on an element in this document.
- `Document.releaseEvents()` {{Deprecated_Inline}}
  - : See {{DOMxRef("Window.releaseEvents()")}}.
- {{DOMxRef("Document.replaceChildren()")}}
  - : Replaces the existing children of a document with a specified new set of children.
- {{DOMxRef("Document.requestStorageAccess()")}}
  - : Allows a document loaded in a third-party context (i.e., embedded in an {{htmlelement("iframe")}}) to request access to unpartitioned cookies, in cases where user agents by default block access to unpartitioned cookies by sites loaded in a third-party context to improve privacy.
- {{DOMxRef("Document.requestStorageAccessFor()")}} {{deprecated_inline}} {{non-standard_inline}}
  - : Allows top-level sites to request third-party cookie access on behalf of embedded content originating from another site in the same related website set.
- {{domxref("Document.startViewTransition()")}}
  - : Starts a new {{domxref("View Transition API", "view transition", "", "nocode")}} and returns a {{domxref("ViewTransition")}} object to represent it.

The `Document` interface is extended with the {{DOMxRef("XPathEvaluator")}} interface:

- {{DOMxRef("Document.createExpression()")}}
  - : Compiles an `XPathExpression` which can then be used for (repeated) evaluations.

- {{DOMxRef("Document.createNSResolver()")}} {{deprecated_inline}}
  - : Returns the input node as-is.
- {{DOMxRef("Document.evaluate()")}}
  - : Evaluates an XPath expression.

Extension for HTML documents

The `Document` interface for HTML documents inherit from the {{DOMxRef("HTMLDocument")}} interface or is extended for such documents:

- {{DOMxRef("Document.clear()")}} {{Deprecated_Inline}}
  - : This method does nothing.
- {{DOMxRef("Document.close()")}}
  - : Closes a document stream for writing.
- {{DOMxRef("Document.execCommand()")}} {{Deprecated_Inline}} {{non-standard_inline}}
  - : On an editable document, executes a formatting command.
- {{DOMxRef("Document.getElementsByName()")}}
  - : Returns a list of elements with the given name.
- {{DOMxRef("Document.hasFocus()")}}
  - : Returns `true` if the focus is currently located anywhere inside the specified document.
- {{DOMxRef("Document.open()")}}
  - : Opens a document stream for writing.
- {{DOMxRef("Document.queryCommandEnabled()")}} {{Deprecated_Inline}} {{Non-standard_Inline}}
  - : Returns true if the formatting command can be executed on the current range.
- `Document.queryCommandIndeterm()` {{Deprecated_Inline}} {{non-standard_inline}}
  - : Returns true if the formatting command is in an indeterminate state on the current range.
- {{DOMxRef("Document.queryCommandState()")}} {{Deprecated_Inline}} {{Non-standard_Inline}}
  - : Returns true if the formatting command has been executed on the current range.
- {{DOMxRef("Document.queryCommandSupported()")}} {{Deprecated_Inline}} {{Non-standard_Inline}}
  - : Returns true if the formatting command is supported on the current range.
- `Document.queryCommandValue()` {{Deprecated_Inline}} {{non-standard_inline}}
  - : Returns the current value of the current range for a formatting command.
- {{DOMxRef("Document.write()")}} {{deprecated_inline}}
  - : Writes text in a document.
- {{DOMxRef("Document.writeln()")}} {{deprecated_inline}}
  - : Writes a line of text in a document.

Static methods

This interface also inherits from the {{DOMxRef("Node")}} and {{DOMxRef("EventTarget")}} interfaces.

- {{domxref("Document/parseHTML_static", "Document.parseHTML()")}}
  - : Creates a new `Document` object from a string of HTML in an XSS-safe manner with sanitization.
- {{domxref("Document/parseHTMLUnsafe_static", "Document.parseHTMLUnsafe()")}}
  - : Creates a new `Document` object from a string of HTML without performing sanitization. The string may contain declarative shadow roots.

Events

Listen to these events using `addEventListener()` or by assigning an event listener to the `oneventname` property of this interface. In addition to the events listed below, many events can bubble from {{domxref("Node", "nodes", "", "nocode")}} contained in the document tree.

- {{DOMxRef("Document.afterscriptexecute_event", "afterscriptexecute")}} {{Non-standard_Inline}} {{deprecated_inline}}
  - : Fired when a static {{HTMLElement("script")}} element finishes executing its script
- {{DOMxRef("Document.beforescriptexecute_event", "beforescriptexecute")}} {{Non-standard_Inline}} {{deprecated_inline}}
  - : Fired when a static {{HTMLElement("script")}} is about to start executing.
- {{domxref("Document.prerenderingchange_event", "prerenderingchange")}} {{experimental_inline}}
  - : Fired on a prerendered document when it is activated (i.e., the user views the page).
- {{DOMxRef("Document.securitypolicyviolation_event", "securitypolicyviolation")}}
  - : Fired when a content security policy is violated.
- {{DOMxRef("Document/visibilitychange_event", "visibilitychange")}}
  - : Fired when the content of a tab has become visible or has been hidden.

Fullscreen events

- {{DOMxRef("Document/fullscreenchange_event", "fullscreenchange")}}
  - : Fired when the `Document` transitions into or out of fullscreen mode.
- {{DOMxRef("Document/fullscreenerror_event", "fullscreenerror")}}
  - : Fired if an error occurs while attempting to switch into or out of fullscreen mode.

Load & unload events

- {{DOMxRef("Document/DOMContentLoaded_event", "DOMContentLoaded")}}
  - : Fired when the document has been completely loaded and parsed, without waiting for stylesheets, images, and subframes to finish loading.
- {{DOMxRef("Document/readystatechange_event", "readystatechange")}}
  - : Fired when the {{DOMxRef("Document/readyState", "readyState")}} attribute of a document has changed.

Pointer lock events

- {{DOMxRef("Document/pointerlockchange_event", "pointerlockchange")}}
  - : Fired when the pointer is locked/unlocked.
- {{DOMxRef("Document/pointerlockerror_event", "pointerlockerror")}}
  - : Fired when locking the pointer failed.

Scroll events

- {{DOMxRef("Document/scroll_event", "scroll")}}
  - : Fired when the document view or an element has been scrolled.
- {{DOMxRef("Document/scrollend_event", "scrollend")}}
  - : Fired when the document view or an element has completed scrolling.
- {{domxref("Document/scrollsnapchange_event", "scrollsnapchange")}} {{experimental_inline}}
  - : Fired on the scroll container at the end of a scrolling operation when a new scroll snap target has been selected.
- {{domxref("Document/scrollsnapchanging_event", "scrollsnapchanging")}} {{experimental_inline}}
  - : Fired on the scroll container when the browser determines a new scroll snap target is pending, i.e., it will be selected when the current scroll gesture ends.

Selection events

- {{DOMxRef("Document/selectionchange_event", "selectionchange")}}
  - : Fired when the current text selection on a document is changed.

Bubbled events

Not all events that bubble can reach the `Document` object. Only the following do and can be listened for on the `Document` object:

- `abort`
- {{domxref("Element/auxclick_event", "auxclick")}}
- {{domxref("Element/beforeinput_event", "beforeinput")}}
- {{domxref("Element/beforematch_event", "beforematch")}}
- {{domxref("HTMLElement/beforetoggle_event", "beforetoggle")}}
- {{domxref("Element/blur_event", "blur")}}
- `cancel`
- {{domxref("HTMLMediaElement/canplay_event", "canplay")}}
- {{domxref("HTMLMediaElement/canplaythrough_event", "canplaythrough")}}
- {{domxref("HTMLElement/change_event", "change")}}
- {{domxref("Element/click_event", "click")}}
- {{domxref("HTMLDialogElement/close_event", "close")}}
- {{domxref("HTMLCanvasElement/contextlost_event", "contextlost")}}
- {{domxref("Element/contextmenu_event", "contextmenu")}}
- {{domxref("HTMLCanvasElement/contextrestored_event", "contextrestored")}}
- {{domxref("Element/copy_event", "copy")}}
- {{domxref("HTMLTrackElement/cuechange_event", "cuechange")}}
- {{domxref("Element/cut_event", "cut")}}
- {{domxref("Element/dblclick_event", "dblclick")}}
- {{domxref("HTMLElement/drag_event", "drag")}}
- {{domxref("HTMLElement/dragend_event", "dragend")}}
- {{domxref("HTMLElement/dragenter_event", "dragenter")}}
- {{domxref("HTMLElement/dragleave_event", "dragleave")}}
- {{domxref("HTMLElement/dragover_event", "dragover")}}
- {{domxref("HTMLElement/dragstart_event", "dragstart")}}
- {{domxref("HTMLElement/drop_event", "drop")}}
- {{domxref("HTMLMediaElement/durationchange_event", "durationchange")}}
- {{domxref("HTMLMediaElement/emptied_event", "emptied")}}
- {{domxref("HTMLMediaElement/ended_event", "ended")}}
- {{domxref("HTMLElement/error_event", "error")}}
- {{domxref("Element/focus_event", "focus")}}
- {{domxref("HTMLFormElement/formdata_event", "formdata")}}
- {{domxref("Element/input_event", "input")}}
- {{domxref("HTMLInputElement/invalid_event", "invalid")}}
- {{domxref("Element/keydown_event", "keydown")}}
- {{domxref("Element/keypress_event", "keypress")}}
- {{domxref("Element/keyup_event", "keyup")}}
- {{domxref("HTMLElement/load_event", "load")}}
- {{domxref("HTMLMediaElement/loadeddata_event", "loadeddata")}}
- {{domxref("HTMLMediaElement/loadedmetadata_event", "loadedmetadata")}}
- {{domxref("HTMLMediaElement/loadstart_event", "loadstart")}}
- {{domxref("Element/mousedown_event", "mousedown")}}
- {{domxref("Element/mouseenter_event", "mouseenter")}}
- {{domxref("Element/mouseleave_event", "mouseleave")}}
- {{domxref("Element/mousemove_event", "mousemove")}}

- {{domxref("Element/mouseout_event", "mouseout")}}
- {{domxref("Element/mouseover_event", "mouseover")}}
- {{domxref("Element/mouseup_event", "mouseup")}}
- {{domxref("Element/paste_event", "paste")}}
- {{domxref("HTMLMediaElement/pause_event", "pause")}}
- {{domxref("HTMLMediaElement/play_event", "play")}}
- {{domxref("HTMLMediaElement/playing_event", "playing")}}
- {{domxref("HTMLMediaElement/progress_event", "progress")}}
- {{domxref("HTMLMediaElement/ratechange_event", "ratechange")}}
- {{domxref("HTMLFormElement/reset_event", "reset")}}
- {{domxref("HTMLVideoElement/resize_event", "resize")}}
- {{domxref("Element/scroll_event", "scroll")}}
- {{domxref("Element/scrollend_event", "scrollend")}}
- {{domxref("Element/securitypolicyviolation_event", "securitypolicyviolation")}}
- {{domxref("HTMLMediaElement/seeked_event", "seeked")}}
- {{domxref("HTMLMediaElement/seeking_event", "seeking")}}
- {{domxref("HTMLInputElement/select_event", "select")}}
- {{domxref("HTMLSlotElement/slotchange_event", "slotchange")}}
- {{domxref("HTMLMediaElement/stalled_event", "stalled")}}
- {{domxref("HTMLFormElement/submit_event", "submit")}}
- {{domxref("HTMLMediaElement/suspend_event", "suspend")}}
- {{domxref("HTMLMediaElement/timeupdate_event", "timeupdate")}}
- {{domxref("HTMLElement/toggle_event", "toggle")}}
- {{domxref("HTMLMediaElement/volumechange_event", "volumechange")}}
- {{domxref("HTMLMediaElement/waiting_event", "waiting")}}
- {{domxref("Element/wheel_event", "wheel")}}

Specifications

{{Specifications}}

Browser compatibility

{{Compat}}

□□□ #2
□ □□□ □□□ 2026-10-31 03:27:53 UTC
□ □□□ □□□ 2026-10-31 03:27:53 UTC