

HTML



Living Standard — Last Updated 21 May 2025

One-Page Version html.spec.whatwg.org	Multipage Version /multipage
Version for Web Devs /dev	PDF Version /print.pdf
Translations 日本語・简体中文	FAQ on GitHub
Chat on Matrix	Contribute on GitHub whatwg/html repository
Commits on GitHub	Snapshot as of this commit
Twitter Updates @htmlstandard	Open Issues filed on GitHub
Open an Issue whatwg.org/newbug	Tests web-platform-tests html/
Issues for Tests ongoing work	

Table of contents

[1 Introduction](#)
[2 Common infrastructure](#)
[3 Semantics, structure, and APIs of HTML documents](#)
[4 The elements of HTML](#)
[5 Microdata](#)
[6 User interaction](#)
[7 Loading web pages](#)
[8 Web application APIs](#)
[9 Communication](#)
[10 Web workers](#)
[11 Worklets](#)
[12 Web storage](#)
[13 The HTML syntax](#)
[14 The XML syntax](#)
[15 Rendering](#)
[16 Obsolete features](#)
[17 IANA considerations](#)
[Index](#)
[References](#)
[Acknowledgments](#)
[Intellectual property rights](#)

Full table of contents

[1 Introduction](#)
[1.1 Where does this specification fit?](#)
[1.2 Is this HTML5?](#)

[1.3 Background](#)[1.4 Audience](#)[1.5 Scope](#)[1.6 History](#)[1.7 Design notes](#)[1.7.1 Serializability of script execution](#)[1.7.2 Extensibility](#)[1.8 HTML vs XML syntax](#)[1.9 Structure of this specification](#)[1.9.1 How to read this specification](#)[1.9.2 Typographic conventions](#)[1.10 A quick introduction to HTML](#)[1.10.1 Writing secure applications with HTML](#)[1.10.2 Common pitfalls to avoid when using the scripting APIs](#)[1.10.3 How to catch mistakes when writing HTML: validators and conformance checkers](#)[1.11 Conformance requirements for authors](#)[1.11.1 Presentational markup](#)[1.11.2 Syntax errors](#)[1.11.3 Restrictions on content models and on attribute values](#)[1.12 Suggested reading](#)[2 Common infrastructure](#)[2.1 Terminology](#)[2.1.1 Parallelism](#)[2.1.2 Resources](#)[2.1.3 XML compatibility](#)[2.1.4 DOM trees](#)[2.1.5 Scripting](#)[2.1.6 Plugins](#)[2.1.7 Character encodings](#)[2.1.8 Conformance classes](#)[2.1.9 Dependencies](#)[2.1.10 Extensibility](#)[2.1.11 Interactions with XPath and XSLT](#)[2.2 Policy-controlled features](#)[2.3 Common microsyntaxes](#)[2.3.1 Common parser idioms](#)[2.3.2 Boolean attributes](#)[2.3.3 Keywords and enumerated attributes](#)[2.3.4 Numbers](#)[2.3.4.1 Signed integers](#)[2.3.4.2 Non-negative integers](#)[2.3.4.3 Floating-point numbers](#)[2.3.4.4 Percentages and lengths](#)[2.3.4.5 Nonzero percentages and lengths](#)[2.3.4.6 Lists of floating-point numbers](#)[2.3.4.7 Lists of dimensions](#)[2.3.5 Dates and times](#)[2.3.5.1 Months](#)[2.3.5.2 Dates](#)

[2.3.5.3 Yearless dates](#)

[2.3.5.4 Times](#)

[2.3.5.5 Local dates and times](#)

[2.3.5.6 Time zones](#)

[2.3.5.7 Global dates and times](#)

[2.3.5.8 Weeks](#)

[2.3.5.9 Durations](#)

[2.3.5.10 Vaguer moments in time](#)

[2.3.6 Legacy colors](#)

[2.3.7 Space-separated tokens](#)

[2.3.8 Comma-separated tokens](#)

[2.3.9 References](#)

[2.3.10 Media queries](#)

[2.3.11 Unique internal values](#)

[2.4 URLs](#)

[2.4.1 Terminology](#)

[2.4.2 Parsing URLs](#)

[2.4.3 Dynamic changes to base URLs](#)

[2.5 Fetching resources](#)

[2.5.1 Terminology](#)

[2.5.2 Determining the type of a resource](#)

[2.5.3 Extracting character encodings from meta elements](#)

[2.5.4 CORS settings attributes](#)

[2.5.5 Referrer policy attributes](#)

[2.5.6 Nonce attributes](#)

[2.5.7 Lazy loading attributes](#)

[2.5.8 Blocking attributes](#)

[2.5.9 Fetch priority attributes](#)

[2.6 Common DOM interfaces](#)

[2.6.1 Reflecting content attributes in IDL attributes](#)

[2.6.2 Using reflect in specifications](#)

[2.6.3 Collections](#)

[2.6.3.1 The `HTMLAllCollection` interface](#)

[2.6.3.1.1 `\[\[Call\]\] \(thisArgument, argumentsList\)`](#)

[2.6.3.2 The `HTMLFormControlsCollection` interface](#)

[2.6.3.3 The `HTMLOptionsCollection` interface](#)

[2.6.4 The `DOMStringList` interface](#)

[2.7 Safe passing of structured data](#)

[2.7.1 Serializable objects](#)

[2.7.2 Transferable objects](#)

[2.7.3 `StructuredSerializeInternal \(value, forStorage \[, memory\]\)`](#)

[2.7.4 `StructuredSerialize \(value\)`](#)

[2.7.5 `StructuredSerializeForStorage \(value\)`](#)

[2.7.6 `StructuredDeserialize \(serialized, targetRealm \[, memory\]\)`](#)

[2.7.7 `StructuredSerializeWithTransfer \(value, transferList\)`](#)

[2.7.8 `StructuredDeserializeWithTransfer \(serializeWithTransferResult, targetRealm\)`](#)

[2.7.9 Performing serialization and transferring from other specifications](#)

[2.7.10 Structured cloning API](#)

[3 Semantics, structure, and APIs of HTML documents](#)

[3.1 Documents](#)

[3.1.1 The `Document` object](#)

[3.1.2 The `DocumentOrShadowRoot` interface](#)

[3.1.3 Resource metadata management](#)

[3.1.4 Reporting document loading status](#)

[3.1.5 Render-blocking mechanism](#)

[3.1.6 DOM tree accessors](#)

[3.2 Elements](#)

[3.2.1 Semantics](#)

[3.2.2 Elements in the DOM](#)

[3.2.3 HTML element constructors](#)

[3.2.4 Element definitions](#)

[3.2.4.1 Attributes](#)

[3.2.5 Content models](#)

[3.2.5.1 The "nothing" content model](#)

[3.2.5.2 Kinds of content](#)

[3.2.5.2.1 Metadata content](#)

[3.2.5.2.2 Flow content](#)

[3.2.5.2.3 Sectioning content](#)

[3.2.5.2.4 Heading content](#)

[3.2.5.2.5 Phrasing content](#)

[3.2.5.2.6 Embedded content](#)

[3.2.5.2.7 Interactive content](#)

[3.2.5.2.8 Palpable content](#)

[3.2.5.2.9 Script-supporting elements](#)

[3.2.5.3 Transparent content models](#)

[3.2.5.4 Paragraphs](#)

[3.2.6 Global attributes](#)

[3.2.6.1 The `title` attribute](#)

[3.2.6.2 The `lang` and `xml:lang` attributes](#)

[3.2.6.3 The `translate` attribute](#)

[3.2.6.4 The `dir` attribute](#)

[3.2.6.5 The `style` attribute](#)

[3.2.6.6 Embedding custom non-visible data with the `data-*` attributes](#)

[3.2.7 The `innerText` and `outerText` properties](#)

[3.2.8 Requirements relating to the bidirectional algorithm](#)

[3.2.8.1 Authoring conformance criteria for bidirectional-algorithm formatting characters](#)

[3.2.8.2 User agent conformance criteria](#)

[3.2.9 Requirements related to ARIA and to platform accessibility APIs](#)

[4 The elements of HTML](#)

[4.1 The document element](#)

[4.1.1 The `html` element](#)

[4.2 Document metadata](#)

[4.2.1 The `head` element](#)

[4.2.2 The `title` element](#)

[4.2.3 The `base` element](#)

[4.2.4 The `link` element](#)

[4.2.4.1 Processing the `media` attribute](#)

[4.2.4.2 Processing the `type` attribute](#)

[4.2.4.3 Fetching and processing a resource from a `link` element](#)

[4.2.4.4 Processing `Link` headers](#)

[4.2.4.5 Early hints](#)

[4.2.4.6 Providing users with a means to follow hyperlinks created using the `link` element](#)

[4.2.5 The `meta` element](#)

[4.2.5.1 Standard metadata names](#)

[4.2.5.2 Other metadata names](#)

[4.2.5.3 Pragma directives](#)

[4.2.5.4 Specifying the document's character encoding](#)

[4.2.6 The `style` element](#)

[4.2.7 Interactions of styling and scripting](#)

[4.3 Sections](#)

[4.3.1 The `body` element](#)

[4.3.2 The `article` element](#)

[4.3.3 The `section` element](#)

[4.3.4 The `nav` element](#)

[4.3.5 The `aside` element](#)

[4.3.6 The `h1`, `h2`, `h3`, `h4`, `h5`, and `h6` elements](#)

[4.3.7 The `hgroup` element](#)

[4.3.8 The `header` element](#)

[4.3.9 The `footer` element](#)

[4.3.10 The `address` element](#)

[4.3.11 Headings and outlines](#)

[4.3.11.1 Sample outlines](#)

[4.3.11.2 Exposing outlines to users](#)

[4.3.12 Usage summary](#)

[4.3.12.1 Article or section?](#)

[4.4 Grouping content](#)

[4.4.1 The `p` element](#)

[4.4.2 The `hr` element](#)

[4.4.3 The `pre` element](#)

[4.4.4 The `blockquote` element](#)

[4.4.5 The `ol` element](#)

[4.4.6 The `ul` element](#)

[4.4.7 The `menu` element](#)

[4.4.8 The `li` element](#)

[4.4.9 The `dl` element](#)

[4.4.10 The `dt` element](#)

[4.4.11 The `dd` element](#)

[4.4.12 The `figure` element](#)

[4.4.13 The `figcaption` element](#)

[4.4.14 The `main` element](#)

[4.4.15 The `search` element](#)

[4.4.16 The `div` element](#)

[4.5 Text-level semantics](#)

[4.5.1 The `a` element](#)

[4.5.2 The `em` element](#)

[4.5.3 The `strong` element](#)

[4.5.4 The `small` element](#)

[4.5.5 The `s` element](#)

[4.5.6 The `cite` element](#)

[4.5.7 The `q` element](#)

[4.5.8 The `dfn` element](#)

[4.5.9 The `abbr` element](#)

[4.5.10 The `ruby` element](#)

[4.5.11 The `rt` element](#)

[4.5.12 The `rp` element](#)

[4.5.13 The `data` element](#)

[4.5.14 The `time` element](#)

[4.5.15 The `code` element](#)

[4.5.16 The `var` element](#)

[4.5.17 The `samp` element](#)

[4.5.18 The `kbd` element](#)

[4.5.19 The `sub` and `sup` elements](#)

[4.5.20 The `i` element](#)

[4.5.21 The `b` element](#)

[4.5.22 The `u` element](#)

[4.5.23 The `mark` element](#)

[4.5.24 The `bdi` element](#)

[4.5.25 The `bdo` element](#)

[4.5.26 The `span` element](#)

[4.5.27 The `br` element](#)

[4.5.28 The `wbr` element](#)

[4.5.29 Usage summary](#)

[4.6 Links](#)

[4.6.1 Introduction](#)

[4.6.2 Links created by `a` and `area` elements](#)

[4.6.3 API for `a` and `area` elements](#)

[4.6.4 Following hyperlinks](#)

[4.6.5 Downloading resources](#)

[4.6.6 Hyperlink auditing](#)

[4.6.6.1 The ``Ping-From`` and ``Ping-To`` headers](#)

[4.6.7 Link types](#)

[4.6.7.1 Link type `"alternate"`](#)

[4.6.7.2 Link type `"author"`](#)

[4.6.7.3 Link type `"bookmark"`](#)

[4.6.7.4 Link type `"canonical"`](#)

[4.6.7.5 Link type `"dns-prefetch"`](#)

[4.6.7.6 Link type `"expect"`](#)

[4.6.7.7 Link type `"external"`](#)

[4.6.7.8 Link type `"help"`](#)

[4.6.7.9 Link type `"icon"`](#)

[4.6.7.10 Link type `"license"`](#)

[4.6.7.11 Link type `"manifest"`](#)

[4.6.7.12 Link type `"modulepreload"`](#)

[4.6.7.13 Link type `"nofollow"`](#)

[4.6.7.14 Link type `"noopener"`](#)

[4.6.7.15 Link type `"noreferrer"`](#)

[4.6.7.16 Link type `"opener"`](#)

[4.6.7.17 Link type `"pingback"`](#)

[4.6.7.18 Link type `"preconnect"`](#)

[4.6.7.19 Link type `"prefetch"`](#)

[4.6.7.20 Link type `"preload"`](#)

[4.6.7.21 Link type `"privacy-policy"`](#)

[4.6.7.22 Link type `"search"`](#)

[4.6.7.23 Link type `"stylesheet"`](#)

[4.6.7.24 Link type `"tag"`](#)

[4.6.7.25 Link Type "terms-of-service"](#)

[4.6.7.26 Sequential link types](#)

[4.6.7.26.1 Link type "next"](#)

[4.6.7.26.2 Link type "prev"](#)

[4.6.7.27 Other link types](#)

[4.7 Edits](#)

[4.7.1 The `ins` element](#)

[4.7.2 The `del` element](#)

[4.7.3 Attributes common to `ins` and `del` elements](#)

[4.7.4 Edits and paragraphs](#)

[4.7.5 Edits and lists](#)

[4.7.6 Edits and tables](#)

[4.8 Embedded content](#)

[4.8.1 The `picture` element](#)

[4.8.2 The `source` element](#)

[4.8.3 The `img` element](#)

[4.8.4 Images](#)

[4.8.4.1 Introduction](#)

[4.8.4.1.1 Adaptive images](#)

[4.8.4.2 Attributes common to `source`, `img`, and `link` elements](#)

[4.8.4.2.1 Srcset attributes](#)

[4.8.4.2.2 Sizes attributes](#)

[4.8.4.3 Processing model](#)

[4.8.4.3.1 When to obtain images](#)

[4.8.4.3.2 Reacting to DOM mutations](#)

[4.8.4.3.3 The list of available images](#)

[4.8.4.3.4 Decoding images](#)

[4.8.4.3.5 Updating the image data](#)

[4.8.4.3.6 Preparing an image for presentation](#)

[4.8.4.3.7 Selecting an image source](#)

[4.8.4.3.8 Creating a source set from attributes](#)

[4.8.4.3.9 Updating the source set](#)

[4.8.4.3.10 Parsing a srcset attribute](#)

[4.8.4.3.11 Parsing a sizes attribute](#)

[4.8.4.3.12 Normalizing the source densities](#)

[4.8.4.3.13 Reacting to environment changes](#)

[4.8.4.4 Requirements for providing text to act as an alternative for images](#)

[4.8.4.4.1 General guidelines](#)

[4.8.4.4.2 A link or button containing nothing but the image](#)

[4.8.4.4.3 A phrase or paragraph with an alternative graphical representation: charts, diagrams, graphs, maps, illustrations](#)

[4.8.4.4.4 A short phrase or label with an alternative graphical representation: icons, logos](#)

[4.8.4.4.5 Text that has been rendered to a graphic for typographical effect](#)

[4.8.4.4.6 A graphical representation of some of the surrounding text](#)

[4.8.4.4.7 Ancillary images](#)

[4.8.4.4.8 A purely decorative image that doesn't add any information](#)

[4.8.4.4.9 A group of images that form a single larger picture with no links](#)

[4.8.4.4.10 A group of images that form a single larger picture with links](#)

[4.8.4.4.11 A key part of the content](#)

[4.8.4.4.12 An image not intended for the user](#)

[4.8.4.4.13 An image in an email or private document intended for a specific person who is known to be able to view images](#)

[4.8.4.4.14 Guidance for markup generators](#)

[4.8.4.4.15 Guidance for conformance checkers](#)

[4.8.5 The `iframe` element](#)

[4.8.6 The `embed` element](#)

[4.8.7 The `object` element](#)

[4.8.8 The `video` element](#)

[4.8.9 The `audio` element](#)

[4.8.10 The **track** element](#)[4.8.11 Media elements](#)[4.8.11.1 Error codes](#)[4.8.11.2 Location of the media resource](#)[4.8.11.3 MIME types](#)[4.8.11.4 Network states](#)[4.8.11.5 Loading the media resource](#)[4.8.11.6 Offsets into the media resource](#)[4.8.11.7 Ready states](#)[4.8.11.8 Playing the media resource](#)[4.8.11.9 Seeking](#)[4.8.11.10 Media resources with multiple media tracks](#)[4.8.11.10.1 **AudioTrackList** and **VideoTrackList** objects](#)[4.8.11.10.2 Selecting specific audio and video tracks declaratively](#)[4.8.11.11 Timed text tracks](#)[4.8.11.11.1 Text track model](#)[4.8.11.11.2 Sourcing in-band text tracks](#)[4.8.11.11.3 Sourcing out-of-band text tracks](#)[4.8.11.11.4 Guidelines for exposing cues in various formats as text track cues](#)[4.8.11.11.5 Text track API](#)[4.8.11.11.6 Event handlers for objects of the text track APIs](#)[4.8.11.11.7 Best practices for metadata text tracks](#)[4.8.11.12 Identifying a track kind through a URL](#)[4.8.11.13 User interface](#)[4.8.11.14 Time ranges](#)[4.8.11.15 The **TrackEvent** interface](#)[4.8.11.16 Events summary](#)[4.8.11.17 Security and privacy considerations](#)[4.8.11.18 Best practices for authors using media elements](#)[4.8.11.19 Best practices for implementers of media elements](#)[4.8.12 The **map** element](#)[4.8.13 The **area** element](#)[4.8.14 Image maps](#)[4.8.14.1 Authoring](#)[4.8.14.2 Processing model](#)[4.8.15 MathML](#)[4.8.16 SVG](#)[4.8.17 Dimension attributes](#)[4.9 Tabular data](#)[4.9.1 The **table** element](#)[4.9.1.1 Techniques for describing tables](#)[4.9.1.2 Techniques for table design](#)[4.9.2 The **caption** element](#)[4.9.3 The **colgroup** element](#)[4.9.4 The **col** element](#)[4.9.5 The **tbody** element](#)[4.9.6 The **thead** element](#)[4.9.7 The **tfoot** element](#)[4.9.8 The **tr** element](#)[4.9.9 The **td** element](#)[4.9.10 The **th** element](#)[4.9.11 Attributes common to **td** and **th** elements](#)[4.9.12 Processing model](#)[4.9.12.1 Forming a table](#)

[4.9.12.2 Forming relationships between data cells and header cells](#)[4.9.13 Examples](#)[4.10 Forms](#)[4.10.1 Introduction](#)[4.10.1.1 Writing a form's user interface](#)[4.10.1.2 Implementing the server-side processing for a form](#)[4.10.1.3 Configuring a form to communicate with a server](#)[4.10.1.4 Client-side form validation](#)[4.10.1.5 Enabling client-side automatic filling of form controls](#)[4.10.1.6 Improving the user experience on mobile devices](#)[4.10.1.7 The difference between the field type, the autofill field name, and the input modality](#)[4.10.1.8 Date, time, and number formats](#)[4.10.2 Categories](#)[4.10.3 The `form` element](#)[4.10.4 The `label` element](#)[4.10.5 The `input` element](#)[4.10.5.1 States of the `type` attribute](#)[4.10.5.1.1 Hidden state \(`type=hidden`\)](#)[4.10.5.1.2 Text \(`type=text`\) state and Search state \(`type=search`\)](#)[4.10.5.1.3 Telephone state \(`type=tel`\)](#)[4.10.5.1.4 URL state \(`type=url`\)](#)[4.10.5.1.5 Email state \(`type=email`\)](#)[4.10.5.1.6 Password state \(`type=password`\)](#)[4.10.5.1.7 Date state \(`type=date`\)](#)[4.10.5.1.8 Month state \(`type=month`\)](#)[4.10.5.1.9 Week state \(`type=week`\)](#)[4.10.5.1.10 Time state \(`type=time`\)](#)[4.10.5.1.11 Local Date and Time state \(`type=datetime-local`\)](#)[4.10.5.1.12 Number state \(`type=number`\)](#)[4.10.5.1.13 Range state \(`type=range`\)](#)[4.10.5.1.14 Color state \(`type=color`\)](#)[4.10.5.1.15 Checkbox state \(`type=checkbox`\)](#)[4.10.5.1.16 Radio Button state \(`type=radio`\)](#)[4.10.5.1.17 File Upload state \(`type=file`\)](#)[4.10.5.1.18 Submit Button state \(`type=submit`\)](#)[4.10.5.1.19 Image Button state \(`type=image`\)](#)[4.10.5.1.20 Reset Button state \(`type=reset`\)](#)[4.10.5.1.21 Button state \(`type=button`\)](#)[4.10.5.2 Implementation notes regarding localization of form controls](#)[4.10.5.3 Common `input` element attributes](#)[4.10.5.3.1 The `maxlength` and `minlength` attributes](#)[4.10.5.3.2 The `size` attribute](#)[4.10.5.3.3 The `readonly` attribute](#)[4.10.5.3.4 The `required` attribute](#)[4.10.5.3.5 The `multiple` attribute](#)[4.10.5.3.6 The `pattern` attribute](#)[4.10.5.3.7 The `min` and `max` attributes](#)[4.10.5.3.8 The `step` attribute](#)[4.10.5.3.9 The `list` attribute](#)[4.10.5.3.10 The `placeholder` attribute](#)[4.10.5.4 Common `input` element APIs](#)[4.10.5.5 Common event behaviors](#)[4.10.6 The `button` element](#)[4.10.7 The `select` element](#)[4.10.8 The `datalist` element](#)[4.10.9 The `optgroup` element](#)[4.10.10 The `option` element](#)[4.10.11 The `textarea` element](#)

- [4.10.12 The **output** element](#)
- [4.10.13 The **progress** element](#)
- [4.10.14 The **meter** element](#)
- [4.10.15 The **fieldset** element](#)
- [4.10.16 The **legend** element](#)
- [4.10.17 Form control infrastructure](#)
 - [4.10.17.1 A form control's value](#)
 - [4.10.17.2 Mutability](#)
 - [4.10.17.3 Association of controls and forms](#)
- [4.10.18 Attributes common to form controls](#)
 - [4.10.18.1 Naming form controls: the **name** attribute](#)
 - [4.10.18.2 Submitting element directionality: the **dirname** attribute](#)
 - [4.10.18.3 Limiting user input length: the **maxlength** attribute](#)
 - [4.10.18.4 Setting minimum input length requirements: the **minlength** attribute](#)
 - [4.10.18.5 Enabling and disabling form controls: the **disabled** attribute](#)
 - [4.10.18.6 Form submission attributes](#)
 - [4.10.18.7 Autofill](#)
 - [4.10.18.7.1 Autofilling form controls: the **autocomplete** attribute](#)
 - [4.10.18.7.2 Processing model](#)
- [4.10.19 APIs for the text control selections](#)
- [4.10.20 Constraints](#)
 - [4.10.20.1 Definitions](#)
 - [4.10.20.2 Constraint validation](#)
 - [4.10.20.3 The constraint validation API](#)
 - [4.10.20.4 Security](#)
- [4.10.21 Form submission](#)
 - [4.10.21.1 Introduction](#)
 - [4.10.21.2 Implicit submission](#)
 - [4.10.21.3 Form submission algorithm](#)
 - [4.10.21.4 Constructing the entry list](#)
 - [4.10.21.5 Selecting a form submission encoding](#)
 - [4.10.21.6 Converting an entry list to a list of name-value pairs](#)
 - [4.10.21.7 URL-encoded form data](#)
 - [4.10.21.8 Multipart form data](#)
 - [4.10.21.9 Plain text form data](#)
 - [4.10.21.10 The **SubmitEvent** interface](#)
 - [4.10.21.11 The **FormDataEvent** interface](#)
- [4.10.22 Resetting a form](#)
- [4.11 Interactive elements](#)
 - [4.11.1 The **details** element](#)
 - [4.11.2 The **summary** element](#)
 - [4.11.3 Commands](#)
 - [4.11.3.1 Facets](#)
 - [4.11.3.2 Using the **a** element to define a command](#)
 - [4.11.3.3 Using the **button** element to define a command](#)
 - [4.11.3.4 Using the **input** element to define a command](#)
 - [4.11.3.5 Using the **option** element to define a command](#)
 - [4.11.3.6 Using the **accesskey** attribute on a **legend** element to define a command](#)
 - [4.11.3.7 Using the **accesskey** attribute to define a command on other elements](#)
 - [4.11.4 The **dialog** element](#)
 - [4.11.5 Dialog light dismiss](#)
- [4.12 Scripting](#)
 - [4.12.1 The **script** element](#)
 - [4.12.1.1 Processing model](#)

[4.12.1.2 Scripting languages](#)

[4.12.1.3 Restrictions for contents of `script` elements](#)

[4.12.1.4 Inline documentation for external scripts](#)

[4.12.1.5 Interaction of `script` elements and XSLT](#)

[4.12.2 The `noscript` element](#)

[4.12.3 The `template` element](#)

[4.12.3.1 Interaction of `template` elements with XSLT and XPath](#)

[4.12.4 The `slot` element](#)

[4.12.5 The `canvas` element](#)

[4.12.5.1 The 2D rendering context](#)

[4.12.5.1.1 Implementation notes](#)

[4.12.5.1.2 The canvas settings](#)

[4.12.5.1.3 The canvas state](#)

[4.12.5.1.4 Line styles](#)

[4.12.5.1.5 Text styles](#)

[4.12.5.1.6 Building paths](#)

[4.12.5.1.7 `Path2D` objects](#)

[4.12.5.1.8 Transformations](#)

[4.12.5.1.9 Image sources for 2D rendering contexts](#)

[4.12.5.1.10 Fill and stroke styles](#)

[4.12.5.1.11 Drawing rectangles to the bitmap](#)

[4.12.5.1.12 Drawing text to the bitmap](#)

[4.12.5.1.13 Drawing paths to the canvas](#)

[4.12.5.1.14 Drawing focus rings](#)

[4.12.5.1.15 Drawing images](#)

[4.12.5.1.16 Pixel manipulation](#)

[4.12.5.1.17 Compositing](#)

[4.12.5.1.18 Image smoothing](#)

[4.12.5.1.19 Shadows](#)

[4.12.5.1.20 Filters](#)

[4.12.5.1.21 Working with externally-defined SVG filters](#)

[4.12.5.1.22 Drawing model](#)

[4.12.5.1.23 Best practices](#)

[4.12.5.1.24 Examples](#)

[4.12.5.2 The `ImageBitmap` rendering context](#)

[4.12.5.2.1 Introduction](#)

[4.12.5.2.2 The `ImageBitmapRenderingContext` interface](#)

[4.12.5.3 The `OffscreenCanvas` interface](#)

[4.12.5.3.1 The offscreen 2D rendering context](#)

[4.12.5.4 Color spaces and color space conversion](#)

[4.12.5.5 Serializing bitmaps to a file](#)

[4.12.5.6 Security with `canvas` elements](#)

[4.12.5.7 Premultiplied alpha and the 2D rendering context](#)

[4.13 Custom elements](#)

[4.13.1 Introduction](#)

[4.13.1.1 Creating an autonomous custom element](#)

[4.13.1.2 Creating a form-associated custom element](#)

[4.13.1.3 Creating a custom element with default accessible roles, states, and properties](#)

[4.13.1.4 Creating a customized built-in element](#)

[4.13.1.5 Drawbacks of autonomous custom elements](#)

[4.13.1.6 Upgrading elements after their creation](#)

[4.13.1.7 Exposing custom element states](#)

[4.13.2 Requirements for custom element constructors and reactions](#)

[4.13.2.1 Preserving custom element state when moved](#)

[4.13.3 Core concepts](#)

[4.13.4 The `CustomElementRegistry` interface](#)

[4.13.5 Upgrades](#)

[4.13.6 Custom element reactions](#)

[4.13.7 Element internals](#)

[4.13.7.1 The `ElementInternals` interface](#)

[4.13.7.2 Shadow root access](#)

[4.13.7.3 Form-associated custom elements](#)

[4.13.7.4 Accessibility semantics](#)

[4.13.7.5 Custom state pseudo-class](#)

[4.14 Common idioms without dedicated elements](#)

[4.14.1 Breadcrumb navigation](#)

[4.14.2 Tag clouds](#)

[4.14.3 Conversations](#)

[4.14.4 Footnotes](#)

[4.15 Disabled elements](#)

[4.16 Matching HTML elements using selectors and CSS](#)

[4.16.1 Case-sensitivity of the CSS `'attr\(\)'` function](#)

[4.16.2 Case-sensitivity of selectors](#)

[4.16.3 Pseudo-classes](#)

[5 Microdata](#)

[5.1 Introduction](#)

[5.1.1 Overview](#)

[5.1.2 The basic syntax](#)

[5.1.3 Typed items](#)

[5.1.4 Global identifiers for items](#)

[5.1.5 Selecting names when defining vocabularies](#)

[5.2 Encoding microdata](#)

[5.2.1 The microdata model](#)

[5.2.2 Items](#)

[5.2.3 Names: the `itemprop` attribute](#)

[5.2.4 Values](#)

[5.2.5 Associating names with items](#)

[5.2.6 Microdata and other namespaces](#)

[5.3 Sample microdata vocabularies](#)

[5.3.1 vCard](#)

[5.3.1.1 Conversion to vCard](#)

[5.3.1.2 Examples](#)

[5.3.2 vEvent](#)

[5.3.2.1 Conversion to iCalendar](#)

[5.3.2.2 Examples](#)

[5.3.3 Licensing works](#)

[5.3.3.1 Examples](#)

[5.4 Converting HTML to other formats](#)

[5.4.1 JSON](#)

[6 User interaction](#)

[6.1 The `hidden` attribute](#)

[6.2 Page visibility](#)

[6.2.1 The `VisibilityStateEntry` interface](#)

[6.3 Inert subtrees](#)

[6.3.1 Modal dialogs and inert subtrees](#)

[6.3.2 The `inert` attribute](#)

[6.4 Tracking user activation](#)

- [6.4.1 Data model](#)
- [6.4.2 Processing model](#)
- [6.4.3 APIs gated by user activation](#)
- [6.4.4 The `UserActivation` interface](#)
- [6.4.5 User agent automation](#)
- [6.5 Activation behavior of elements](#)
 - [6.5.1 The `ToggleEvent` interface](#)
 - [6.5.2 The `CommandEvent` interface](#)
- [6.6 Focus](#)
 - [6.6.1 Introduction](#)
 - [6.6.2 Data model](#)
 - [6.6.3 The `tabindex` attribute](#)
 - [6.6.4 Processing model](#)
 - [6.6.5 Sequential focus navigation](#)
 - [6.6.6 Focus management APIs](#)
 - [6.6.7 The `autofocus` attribute](#)
- [6.7 Assigning keyboard shortcuts](#)
 - [6.7.1 Introduction](#)
 - [6.7.2 The `accesskey` attribute](#)
 - [6.7.3 Processing model](#)
- [6.8 Editing](#)
 - [6.8.1 Making document regions editable: The `contenteditable` content attribute](#)
 - [6.8.2 Making entire documents editable: the `designMode` getter and setter](#)
 - [6.8.3 Best practices for in-page editors](#)
 - [6.8.4 Editing APIs](#)
 - [6.8.5 Spelling and grammar checking](#)
 - [6.8.6 Writing suggestions](#)
 - [6.8.7 Autocapitalization](#)
 - [6.8.8 Autocorrection](#)
 - [6.8.9 Input modalities: the `inputmode` attribute](#)
 - [6.8.10 Input modalities: the `enterkeyhint` attribute](#)
- [6.9 Find-in-page](#)
 - [6.9.1 Introduction](#)
 - [6.9.2 Interaction with `details` and `hidden=until-found`](#)
 - [6.9.3 Interaction with selection](#)
- [6.10 Close requests and close watchers](#)
 - [6.10.1 Close requests](#)
 - [6.10.2 Close watcher infrastructure](#)
 - [6.10.3 The `CloseWatcher` interface](#)
- [6.11 Drag and drop](#)
 - [6.11.1 Introduction](#)
 - [6.11.2 The drag data store](#)
 - [6.11.3 The `DataTransfer` interface](#)
 - [6.11.3.1 The `DataTransferItemList` interface](#)
 - [6.11.3.2 The `DataTransferItem` interface](#)
 - [6.11.4 The `DragEvent` interface](#)
 - [6.11.5 Processing model](#)
 - [6.11.6 Events summary](#)
 - [6.11.7 The `draggable` attribute](#)
 - [6.11.8 Security risks in the drag-and-drop model](#)

[6.12 The **popover** attribute](#)[6.12.1 The **popover** target attributes](#)[6.12.2 Popover light dismiss](#)[7 Loading web pages](#)[7.1 Supporting concepts](#)[7.1.1 Origins](#)[7.1.1.1 Sites](#)[7.1.1.2 Relaxing the same-origin restriction](#)[7.1.2 Origin-keyed agent clusters](#)[7.1.3 Cross-origin opener policies](#)[7.1.3.1 The headers](#)[7.1.3.2 Browsing context group switches due to opener policy](#)[7.1.3.3 Reporting](#)[7.1.4 Cross-origin embedder policies](#)[7.1.4.1 The headers](#)[7.1.4.2 Embedder policy checks](#)[7.1.5 Sandboxing](#)[7.1.6 Policy containers](#)[7.2 APIs related to navigation and session history](#)[7.2.1 Security infrastructure for **Window**, **WindowProxy**, and **Location** objects](#)[7.2.1.1 Integration with IDL](#)[7.2.1.2 Shared internal slot: `\[\[CrossOriginPropertyDescriptorMap\]\]`](#)[7.2.1.3 Shared abstract operations](#)[7.2.1.3.1 `CrossOriginProperties` \(*O* \)](#)[7.2.1.3.2 `CrossOriginPropertyFallback` \(*P* \)](#)[7.2.1.3.3 `IsPlatformObjectSameOrigin` \(*O* \)](#)[7.2.1.3.4 `CrossOriginGetOwnPropertyHelper` \(*O*, *P* \)](#)[7.2.1.3.5 `CrossOriginGet` \(*O*, *P*, *Receiver* \)](#)[7.2.1.3.6 `CrossOriginSet` \(*O*, *P*, *V*, *Receiver* \)](#)[7.2.1.3.7 `CrossOriginOwnPropertyKeys` \(*O* \)](#)[7.2.2 The **Window** object](#)[7.2.2.1 Opening and closing windows](#)[7.2.2.2 Indexed access on the **Window** object](#)[7.2.2.3 Named access on the **Window** object](#)[7.2.2.4 Accessing related windows](#)[7.2.2.5 Historical browser interface element APIs](#)[7.2.2.6 Script settings for **Window** objects](#)[7.2.3 The **WindowProxy** exotic object](#)[7.2.3.1 `\[\[GetPrototypeOf\]\]` \(*.* \)](#)[7.2.3.2 `\[\[SetPrototypeOf\]\]` \(*V* \)](#)[7.2.3.3 `\[\[IsExtensible\]\]` \(*.* \)](#)[7.2.3.4 `\[\[PreventExtensions\]\]` \(*.* \)](#)[7.2.3.5 `\[\[GetOwnProperty\]\]` \(*P* \)](#)[7.2.3.6 `\[\[DefineOwnProperty\]\]` \(*P*, *Desc* \)](#)[7.2.3.7 `\[\[Get\]\]` \(*P*, *Receiver* \)](#)[7.2.3.8 `\[\[Set\]\]` \(*P*, *V*, *Receiver* \)](#)[7.2.3.9 `\[\[Delete\]\]` \(*P* \)](#)[7.2.3.10 `\[\[OwnPropertyKeys\]\]` \(*.* \)](#)[7.2.4 The **Location** interface](#)[7.2.4.1 `\[\[GetPrototypeOf\]\]` \(*.* \)](#)[7.2.4.2 `\[\[SetPrototypeOf\]\]` \(*V* \)](#)[7.2.4.3 `\[\[IsExtensible\]\]` \(*.* \)](#)[7.2.4.4 `\[\[PreventExtensions\]\]` \(*.* \)](#)[7.2.4.5 `\[\[GetOwnProperty\]\]` \(*P* \)](#)

[7.2.4.6 \[\[DefineOwnProperty\]\]\(.P, Desc\)](#)

[7.2.4.7 \[\[Get\]\]\(.P, Receiver\)](#)

[7.2.4.8 \[\[Set\]\]\(.P, V, Receiver\)](#)

[7.2.4.9 \[\[Delete\]\]\(.P\)](#)

[7.2.4.10 \[\[OwnPropertyKeys\]\]\(\)](#)

[7.2.5 The History interface](#)

[7.2.6 The navigation API](#)

[7.2.6.1 Introduction](#)

[7.2.6.2 The Navigation interface](#)

[7.2.6.3 Core infrastructure](#)

[7.2.6.4 Initializing and updating the entry list](#)

[7.2.6.5 The NavigationHistoryEntry interface](#)

[7.2.6.6 The history entry list](#)

[7.2.6.7 Initiating navigations](#)

[7.2.6.8 Ongoing navigation tracking](#)

[7.2.6.9 The NavigationActivation interface](#)

[7.2.6.10 The navigate event](#)

[7.2.6.10.1 The NavigateEvent interface](#)

[7.2.6.10.2 The NavigationDestination interface](#)

[7.2.6.10.3 Firing the event](#)

[7.2.6.10.4 Scroll and focus behavior](#)

[7.2.7 Event interfaces](#)

[7.2.7.1 The NavigationCurrentEntryChangeEvent interface](#)

[7.2.7.2 The PopStateEvent interface](#)

[7.2.7.3 The HashChangeEvent interface](#)

[7.2.7.4 The PageSwapEvent interface](#)

[7.2.7.5 The PageRevealEvent interface](#)

[7.2.7.6 The PageTransitionEvent interface](#)

[7.2.7.7 The BeforeUnloadEvent interface](#)

[7.2.8 The NotRestoredReasons interface](#)

[7.3 Infrastructure for sequences of documents](#)

[7.3.1 Navigables](#)

[7.3.1.1 Traversable navigables](#)

[7.3.1.2 Top-level traversables](#)

[7.3.1.3 Child navigables](#)

[7.3.1.4 Jake diagrams](#)

[7.3.1.5 Related navigable collections](#)

[7.3.1.6 Navigable destruction](#)

[7.3.1.7 Navigable target names](#)

[7.3.2 Browsing contexts](#)

[7.3.2.1 Creating browsing contexts](#)

[7.3.2.2 Related browsing contexts](#)

[7.3.2.3 Groupings of browsing contexts](#)

[7.3.3 Fully active documents](#)

[7.4 Navigation and session history](#)

[7.4.1 Session history](#)

[7.4.1.1 Session history entries](#)

[7.4.1.2 Document state](#)

[7.4.1.3 Centralized modifications of session history](#)

[7.4.1.4 Low-level operations on session history](#)

[7.4.2 Navigation](#)

[7.4.2.1 Supporting concepts](#)

[7.4.2.2 Beginning navigation](#)

[7.4.2.3 Ending navigation](#)

[7.4.2.3.1 The usual cross-document navigation case](#)

[7.4.2.3.2 The `javascript:` URL special case](#)

[7.4.2.3.3 Fragment navigations](#)

[7.4.2.3.4 Non-fetch schemes and external software](#)

[7.4.2.4 Preventing navigation](#)

[7.4.2.5 Aborting navigation](#)

[7.4.3 Reloading and traversing](#)

[7.4.4 Non-fragment synchronous "navigations"](#)

[7.4.5 Populating a session history entry](#)

[7.4.6 Applying the history step](#)

[7.4.6.1 Updating the traversable](#)

[7.4.6.2 Updating the document](#)

[7.4.6.3 Revealing the document](#)

[7.4.6.4 Scrolling to a fragment](#)

[7.4.6.5 Persisted history entry state](#)

[7.5 Document lifecycle](#)

[7.5.1 Shared document creation infrastructure](#)

[7.5.2 Loading HTML documents](#)

[7.5.3 Loading XML documents](#)

[7.5.4 Loading text documents](#)

[7.5.5 Loading `multipart/x-mixed-replace` documents](#)

[7.5.6 Loading media documents](#)

[7.5.7 Loading a document for inline content that doesn't have a DOM](#)

[7.5.8 Finishing the loading process](#)

[7.5.9 Unloading documents](#)

[7.5.10 Destroying documents](#)

[7.5.11 Aborting a document load](#)

[7.6 The ``X-Frame-Options`` header](#)

[7.7 The ``Refresh`` header](#)

[7.8 Browser user interface considerations](#)

[8 Web application APIs](#)

[8.1 Scripting](#)

[8.1.1 Introduction](#)

[8.1.2 Agents and agent clusters](#)

[8.1.2.1 Integration with the JavaScript agent formalism](#)

[8.1.2.2 Integration with the JavaScript agent cluster formalism](#)

[8.1.3 Realms and their counterparts](#)

[8.1.3.1 Environments](#)

[8.1.3.2 Environment settings objects](#)

[8.1.3.3 Realms, settings objects, and global objects](#)

[8.1.3.3.1 Entry](#)

[8.1.3.3.2 Incumbent](#)

[8.1.3.3.3 Current](#)

[8.1.3.3.4 Relevant](#)

[8.1.3.4 Enabling and disabling scripting](#)

[8.1.3.5 Secure contexts](#)

[8.1.4 Script processing model](#)

[8.1.4.1 Scripts](#)

[8.1.4.2 Fetching scripts](#)

[8.1.4.3 Creating scripts](#)

[8.1.4.4 Calling scripts](#)

[8.1.4.5 Killing scripts](#)

[8.1.4.6 Runtime script errors](#)

- [8.1.4.7 Unhandled promise rejections](#)
- [8.1.4.8 Import map parse results](#)
- [8.1.5 Module specifier resolution](#)
 - [8.1.5.1 The resolution algorithm](#)
 - [8.1.5.2 Import maps](#)
 - [8.1.5.3 Import map processing model](#)
- [8.1.6 JavaScript specification host hooks](#)
 - [8.1.6.1 HostEnsureCanAddPrivateElement\(*O*\)](#)
 - [8.1.6.2 HostEnsureCanCompileStrings\(*realm*, *parameterStrings*, *bodyString*, *codeString*, *compilationType*, *parameterArgs*, *bodyArg*\)](#)
 - [8.1.6.3 HostGetCodeForEval\(*argument*\)](#)
 - [8.1.6.4 HostPromiseRejectionTracker\(*promise*, *operation*\)](#)
 - [8.1.6.5 HostSystemUTCEpochNanoseconds\(*global*\)](#)
 - [8.1.6.6 Job-related host hooks](#)
 - [8.1.6.6.1 HostCallJobCallback\(*callback*, *V*, *argumentsList*\)](#)
 - [8.1.6.6.2 HostEnqueueFinalizationRegistryCleanupJob\(*finalizationRegistry*\)](#)
 - [8.1.6.6.3 HostEnqueueGenericJob\(*job*, *realm*\)](#)
 - [8.1.6.6.4 HostEnqueuePromiseJob\(*job*, *realm*\)](#)
 - [8.1.6.6.5 HostEnqueueTimeoutJob\(*job*, *realm*, *milliseconds*\)](#)
 - [8.1.6.6.6 HostMakeJobCallback\(*callable*\)](#)
 - [8.1.6.7 Module-related host hooks](#)
 - [8.1.6.7.1 HostGetImportMetaProperties\(*moduleRecord*\)](#)
 - [8.1.6.7.2 HostGetSupportedImportAttributes\(\)](#)
 - [8.1.6.7.3 HostLoadImportedModule\(*referrer*, *moduleRequest*, *loadState*, *payload*\)](#)
- [8.1.7 Event loops](#)
 - [8.1.7.1 Definitions](#)
 - [8.1.7.2 Queuing tasks](#)
 - [8.1.7.3 Processing model](#)
 - [8.1.7.4 Generic task sources](#)
 - [8.1.7.5 Dealing with the event loop from other specifications](#)
- [8.1.8 Events](#)
 - [8.1.8.1 Event handlers](#)
 - [8.1.8.2 Event handlers on elements, `Document` objects, and `Window` objects](#)
 - [8.1.8.2.1 IDL definitions](#)
 - [8.1.8.3 Event firing](#)
- [8.2 The `WindowOrWorkerGlobalScope` mixin](#)
- [8.3 Base64 utility methods](#)
- [8.4 Dynamic markup insertion](#)
 - [8.4.1 Opening the input stream](#)
 - [8.4.2 Closing the input stream](#)
 - [8.4.3 `document.write\(\)`](#)
 - [8.4.4 `document.writeln\(\)`](#)
- [8.5 DOM parsing and serialization APIs](#)
 - [8.5.1 The `DOMParser` interface](#)
 - [8.5.2 Unsafe HTML parsing methods](#)
 - [8.5.3 HTML serialization methods](#)
 - [8.5.4 The `innerHTML` property](#)
 - [8.5.5 The `outerHTML` property](#)
 - [8.5.6 The `insertAdjacentHTML\(\)` method](#)
 - [8.5.7 The `createContextualFragment\(\)` method](#)
 - [8.5.8 The `XMLSerializer` interface](#)
- [8.6 Timers](#)
- [8.7 Microtask queuing](#)
- [8.8 User prompts](#)

[8.8.1 Simple dialogs](#)

[8.8.2 Printing](#)

[8.9 System state and capabilities](#)

[8.9.1 The **Navigator** object](#)

[8.9.1.1 Client identification](#)

[8.9.1.2 Language preferences](#)

[8.9.1.3 Browser state](#)

[8.9.1.4 Custom scheme handlers: the **registerProtocolHandler\(\)** method](#)

[8.9.1.4.1 Security and privacy](#)

[8.9.1.4.2 User agent automation](#)

[8.9.1.5 Cookies](#)

[8.9.1.6 PDF viewing support](#)

[8.10 Images](#)

[8.10.1 The **ImageData** interface](#)

[8.10.2 The **ImageBitmap** interface](#)

[8.11 Animation frames](#)

[9 Communication](#)

[9.1 The **MessageEvent** interface](#)

[9.2 Server-sent events](#)

[9.2.1 Introduction](#)

[9.2.2 The **EventSource** interface](#)

[9.2.3 Processing model](#)

[9.2.4 The **`Last-Event-ID`** header](#)

[9.2.5 Parsing an event stream](#)

[9.2.6 Interpreting an event stream](#)

[9.2.7 Authoring notes](#)

[9.2.8 Connectionless push and other features](#)

[9.2.9 Garbage collection](#)

[9.2.10 Implementation advice](#)

[9.3 Cross-document messaging](#)

[9.3.1 Introduction](#)

[9.3.2 Security](#)

[9.3.2.1 Authors](#)

[9.3.2.2 User agents](#)

[9.3.3 Posting messages](#)

[9.4 Channel messaging](#)

[9.4.1 Introduction](#)

[9.4.1.1 Examples](#)

[9.4.1.2 Ports as the basis of an object-capability model on the web](#)

[9.4.1.3 Ports as the basis of abstracting out service implementations](#)

[9.4.2 Message channels](#)

[9.4.3 The **MessageEventTarget** mixin](#)

[9.4.4 Message ports](#)

[9.4.5 Ports and garbage collection](#)

[9.5 Broadcasting to other browsing contexts](#)

[10 Web workers](#)

[10.1 Introduction](#)

[10.1.1 Scope](#)

[10.1.2 Examples](#)

[10.1.2.1 A background number-crunching worker](#)

[10.1.2.2 Using a JavaScript module as a worker](#)

[10.1.2.3 Shared workers introduction](#)

[10.1.2.4 Shared state using a shared worker](#)

[10.1.2.5 Delegation](#)

[10.1.2.6 Providing libraries](#)

[10.1.3 Tutorials](#)

[10.1.3.1 Creating a dedicated worker](#)

[10.1.3.2 Communicating with a dedicated worker](#)

[10.1.3.3 Shared workers](#)

[10.2 Infrastructure](#)

[10.2.1 The global scope](#)

[10.2.1.1 The `WorkerGlobalScope` common interface](#)

[10.2.1.2 Dedicated workers and the `DedicatedWorkerGlobalScope` interface](#)

[10.2.1.3 Shared workers and the `SharedWorkerGlobalScope` interface](#)

[10.2.2 The event loop](#)

[10.2.3 The worker's lifetime](#)

[10.2.4 Processing model](#)

[10.2.5 Runtime script errors](#)

[10.2.6 Creating workers](#)

[10.2.6.1 The `AbstractWorker` mixin](#)

[10.2.6.2 Script settings for workers](#)

[10.2.6.3 Dedicated workers and the `Worker` interface](#)

[10.2.6.4 Shared workers and the `SharedWorker` interface](#)

[10.2.7 Concurrent hardware capabilities](#)

[10.3 APIs available to workers](#)

[10.3.1 Importing scripts and libraries](#)

[10.3.2 The `WorkerNavigator` interface](#)

[10.3.3 The `WorkerLocation` interface](#)

[11 Worklets](#)

[11.1 Introduction](#)

[11.1.1 Motivations](#)

[11.1.2 Code idempotence](#)

[11.1.3 Speculative evaluation](#)

[11.2 Examples](#)

[11.2.1 Loading scripts](#)

[11.2.2 Registering a class and invoking its methods](#)

[11.3 Infrastructure](#)

[11.3.1 The global scope](#)

[11.3.1.1 Agents and event loops](#)

[11.3.1.2 Creation and termination](#)

[11.3.1.3 Script settings for worklets](#)

[11.3.2 The `Worklet` class](#)

[11.3.3 The worklet's lifetime](#)

[12 Web storage](#)

[12.1 Introduction](#)

[12.2 The API](#)

[12.2.1 The `Storage` interface](#)

[12.2.2 The `sessionStorage` getter](#)

[12.2.3 The `localStorage` getter](#)

[12.2.4 The **StorageEvent** interface](#)[12.3 Privacy](#)[12.3.1 User tracking](#)[12.3.2 Sensitivity of data](#)[12.4 Security](#)[12.4.1 DNS spoofing attacks](#)[12.4.2 Cross-directory attacks](#)[12.4.3 Implementation risks](#)[13 The HTML syntax](#)[13.1 Writing HTML documents](#)[13.1.1 The DOCTYPE](#)[13.1.2 Elements](#)[13.1.2.1 Start tags](#)[13.1.2.2 End tags](#)[13.1.2.3 Attributes](#)[13.1.2.4 Optional tags](#)[13.1.2.5 Restrictions on content models](#)[13.1.2.6 Restrictions on the contents of raw text and escapable raw text elements](#)[13.1.3 Text](#)[13.1.3.1 Newlines](#)[13.1.4 Character references](#)[13.1.5 CDATA sections](#)[13.1.6 Comments](#)[13.2 Parsing HTML documents](#)[13.2.1 Overview of the parsing model](#)[13.2.2 Parse errors](#)[13.2.3 The input byte stream](#)[13.2.3.1 Parsing with a known character encoding](#)[13.2.3.2 Determining the character encoding](#)[13.2.3.3 Character encodings](#)[13.2.3.4 Changing the encoding while parsing](#)[13.2.3.5 Preprocessing the input stream](#)[13.2.4 Parse state](#)[13.2.4.1 The insertion mode](#)[13.2.4.2 The stack of open elements](#)[13.2.4.3 The list of active formatting elements](#)[13.2.4.4 The element pointers](#)[13.2.4.5 Other parsing state flags](#)[13.2.5 Tokenization](#)[13.2.5.1 Data state](#)[13.2.5.2 RCDATA state](#)[13.2.5.3 RAWTEXT state](#)[13.2.5.4 Script data state](#)[13.2.5.5 PLAINTEXT state](#)[13.2.5.6 Tag open state](#)[13.2.5.7 End tag open state](#)[13.2.5.8 Tag name state](#)[13.2.5.9 RCDATA less-than sign state](#)[13.2.5.10 RCDATA end tag open state](#)[13.2.5.11 RCDATA end tag name state](#)[13.2.5.12 RAWTEXT less-than sign state](#)[13.2.5.13 RAWTEXT end tag open state](#)

[13.2.5.14 RAWTEXT end tag name state](#)
[13.2.5.15 Script data less-than sign state](#)
[13.2.5.16 Script data end tag open state](#)
[13.2.5.17 Script data end tag name state](#)
[13.2.5.18 Script data escape start state](#)
[13.2.5.19 Script data escape start dash state](#)
[13.2.5.20 Script data escaped state](#)
[13.2.5.21 Script data escaped dash state](#)
[13.2.5.22 Script data escaped dash dash state](#)
[13.2.5.23 Script data escaped less-than sign state](#)
[13.2.5.24 Script data escaped end tag open state](#)
[13.2.5.25 Script data escaped end tag name state](#)
[13.2.5.26 Script data double escape start state](#)
[13.2.5.27 Script data double escaped state](#)
[13.2.5.28 Script data double escaped dash state](#)
[13.2.5.29 Script data double escaped dash dash state](#)
[13.2.5.30 Script data double escaped less-than sign state](#)
[13.2.5.31 Script data double escape end state](#)
[13.2.5.32 Before attribute name state](#)
[13.2.5.33 Attribute name state](#)
[13.2.5.34 After attribute name state](#)
[13.2.5.35 Before attribute value state](#)
[13.2.5.36 Attribute value \(double-quoted\) state](#)
[13.2.5.37 Attribute value \(single-quoted\) state](#)
[13.2.5.38 Attribute value \(unquoted\) state](#)
[13.2.5.39 After attribute value \(quoted\) state](#)
[13.2.5.40 Self-closing start tag state](#)
[13.2.5.41 Bogus comment state](#)
[13.2.5.42 Markup declaration open state](#)
[13.2.5.43 Comment start state](#)
[13.2.5.44 Comment start dash state](#)
[13.2.5.45 Comment state](#)
[13.2.5.46 Comment less-than sign state](#)
[13.2.5.47 Comment less-than sign bang state](#)
[13.2.5.48 Comment less-than sign bang dash state](#)
[13.2.5.49 Comment less-than sign bang dash dash state](#)
[13.2.5.50 Comment end dash state](#)
[13.2.5.51 Comment end state](#)
[13.2.5.52 Comment end bang state](#)
[13.2.5.53 DOCTYPE state](#)
[13.2.5.54 Before DOCTYPE name state](#)
[13.2.5.55 DOCTYPE name state](#)
[13.2.5.56 After DOCTYPE name state](#)
[13.2.5.57 After DOCTYPE public keyword state](#)
[13.2.5.58 Before DOCTYPE public identifier state](#)
[13.2.5.59 DOCTYPE public identifier \(double-quoted\) state](#)
[13.2.5.60 DOCTYPE public identifier \(single-quoted\) state](#)
[13.2.5.61 After DOCTYPE public identifier state](#)
[13.2.5.62 Between DOCTYPE public and system identifiers state](#)
[13.2.5.63 After DOCTYPE system keyword state](#)
[13.2.5.64 Before DOCTYPE system identifier state](#)
[13.2.5.65 DOCTYPE system identifier \(double-quoted\) state](#)
[13.2.5.66 DOCTYPE system identifier \(single-quoted\) state](#)
[13.2.5.67 After DOCTYPE system identifier state](#)
[13.2.5.68 Bogus DOCTYPE state](#)

- [13.2.5.69 CDATA section state](#)
- [13.2.5.70 CDATA section bracket state](#)
- [13.2.5.71 CDATA section end state](#)
- [13.2.5.72 Character reference state](#)
- [13.2.5.73 Named character reference state](#)
- [13.2.5.74 Ambiguous ampersand state](#)
- [13.2.5.75 Numeric character reference state](#)
- [13.2.5.76 Hexadecimal character reference start state](#)
- [13.2.5.77 Decimal character reference start state](#)
- [13.2.5.78 Hexadecimal character reference state](#)
- [13.2.5.79 Decimal character reference state](#)
- [13.2.5.80 Numeric character reference end state](#)

[13.2.6 Tree construction](#)

- [13.2.6.1 Creating and inserting nodes](#)
- [13.2.6.2 Parsing elements that contain only text](#)
- [13.2.6.3 Closing elements that have implied end tags](#)
- [13.2.6.4 The rules for parsing tokens in HTML content](#)
 - [13.2.6.4.1 The "initial" insertion mode](#)
 - [13.2.6.4.2 The "before html" insertion mode](#)
 - [13.2.6.4.3 The "before head" insertion mode](#)
 - [13.2.6.4.4 The "in head" insertion mode](#)
 - [13.2.6.4.5 The "in head noscript" insertion mode](#)
 - [13.2.6.4.6 The "after head" insertion mode](#)
 - [13.2.6.4.7 The "in body" insertion mode](#)
 - [13.2.6.4.8 The "text" insertion mode](#)
 - [13.2.6.4.9 The "in table" insertion mode](#)
 - [13.2.6.4.10 The "in table text" insertion mode](#)
 - [13.2.6.4.11 The "in caption" insertion mode](#)
 - [13.2.6.4.12 The "in column group" insertion mode](#)
 - [13.2.6.4.13 The "in table body" insertion mode](#)
 - [13.2.6.4.14 The "in row" insertion mode](#)
 - [13.2.6.4.15 The "in cell" insertion mode](#)
 - [13.2.6.4.16 The "in select" insertion mode](#)
 - [13.2.6.4.17 The "in select in table" insertion mode](#)
 - [13.2.6.4.18 The "in template" insertion mode](#)
 - [13.2.6.4.19 The "after body" insertion mode](#)
 - [13.2.6.4.20 The "in frameset" insertion mode](#)
 - [13.2.6.4.21 The "after frameset" insertion mode](#)
 - [13.2.6.4.22 The "after after body" insertion mode](#)
 - [13.2.6.4.23 The "after after frameset" insertion mode](#)

[13.2.6.5 The rules for parsing tokens in foreign content](#)

[13.2.7 The end](#)

[13.2.8 Speculative HTML parsing](#)

[13.2.9 Coercing an HTML DOM into an infoset](#)

[13.2.10 An introduction to error handling and strange cases in the parser](#)

- [13.2.10.1 Misnested tags: <i></i>](#)
- [13.2.10.2 Misnested tags: <p></p>](#)
- [13.2.10.3 Unexpected markup in tables](#)
- [13.2.10.4 Scripts that modify the page as it is being parsed](#)
- [13.2.10.5 The execution of scripts that are moving across multiple documents](#)
- [13.2.10.6 Unclosed formatting elements](#)

[13.3 Serializing HTML fragments](#)

[13.4 Parsing HTML fragments](#)

[13.5 Named character references](#)

[14 The XML syntax](#)

[14.1 Writing documents in the XML syntax](#)

[14.2 Parsing XML documents](#)

[14.3 Serializing XML fragments](#)

[14.4 Parsing XML fragments](#)

[15 Rendering](#)

[15.1 Introduction](#)

[15.2 The CSS user agent style sheet and presentational hints](#)

[15.3 Non-replaced elements](#)

[15.3.1 Hidden elements](#)

[15.3.2 The `page`](#)

[15.3.3 Flow content](#)

[15.3.4 Phrasing content](#)

[15.3.5 Bidirectional text](#)

[15.3.6 Sections and headings](#)

[15.3.7 Lists](#)

[15.3.8 Tables](#)

[15.3.9 Margin collapsing quirks](#)

[15.3.10 Form controls](#)

[15.3.11 The `hr` element](#)

[15.3.12 The `fieldset` and `legend` elements](#)

[15.4 Replaced elements](#)

[15.4.1 Embedded content](#)

[15.4.2 Images](#)

[15.4.3 Attributes for embedded content and images](#)

[15.4.4 Image maps](#)

[15.5 Widgets](#)

[15.5.1 Native appearance](#)

[15.5.2 Writing mode](#)

[15.5.3 Button layout](#)

[15.5.4 The `button` element](#)

[15.5.5 The `details` and `summary` elements](#)

[15.5.6 The `input` element as a text entry widget](#)

[15.5.7 The `input` element as domain-specific widgets](#)

[15.5.8 The `input` element as a range control](#)

[15.5.9 The `input` element as a color well](#)

[15.5.10 The `input` element as a checkbox and radio button widgets](#)

[15.5.11 The `input` element as a file upload control](#)

[15.5.12 The `input` element as a button](#)

[15.5.13 The `marquee` element](#)

[15.5.14 The `meter` element](#)

[15.5.15 The `progress` element](#)

[15.5.16 The `select` element](#)

[15.5.17 The `textarea` element](#)

[15.6 Frames and framesets](#)

[15.7 Interactive media](#)

[15.7.1 Links, forms, and navigation](#)

[15.7.2 The `title` attribute](#)

[15.7.3 Editing hosts](#)

[15.7.4 Text rendered in native user interfaces](#)

[15.8 Print media](#)

[15.9 Unstyled XML documents](#)

[16 Obsolete features](#)

[16.1 Obsolete but conforming features](#)

[16.1.1 Warnings for obsolete but conforming features](#)

[16.2 Non-conforming features](#)

[16.3 Requirements for implementations](#)

[16.3.1 The `marquee` element](#)

[16.3.2 Frames](#)

[16.3.3 Other elements, attributes and APIs](#)

[17 IANA considerations](#)

[17.1 `text/html`](#)

[17.2 `multipart/x-mixed-replace`](#)

[17.3 `application/xhtml+xml`](#)

[17.4 `text/ping`](#)

[17.5 `application/microdata+json`](#)

[17.6 `text/event-stream`](#)

[17.7 `web+` scheme prefix](#)

[Index](#)

[Elements](#)

[Element content categories](#)

[Attributes](#)

[Element interfaces](#)

[All interfaces](#)

[Events](#)

[HTTP headers](#)

[MIME types](#)

[References](#)

[Acknowledgments](#)

[Intellectual property rights](#)