Some notes on HTML

John R Hudson[*]

27th October 2023

Contents

[*] The author would welcome notification of any errors or possible misunderstandings.

[1] Table 1 on page 9 offers a simple template for an HTML page. For an evolving template, go to the HTML5 Boilerplate.

1 Introduction

These notes are taken from *HTML Up and Running* (Pilgrim, 2010), *HTML: living standard* (Hickson, 2011, 2013, 2014, 2015, 2018, 2020, 2023) and *Accessible Rich Internet Applications (WAI-ARIA) 1.2*. They are intended as a quick reference for designing web pages and not as a replacement for reading the source documentation which is regularly updated.

Note that:

- you don't have to change anything you have written in the past to get HTML5[2] to work;
- any element or attribute you may have seen previously which is *not* listed is likely to be obsolete;
- HTML5 offers significant enhancements to tablet and smartphone users' experience without sacrificing compatibility with older browsers
- The development of HTML and CSS summarises the key changes since 2011.

This document also includes the recommended ARIA assistive technology attributes for each element; these are described in section 18.2 on page 60.

1.1 XHTML5

Much of this document will also be of assistance to those using XHTML5 provided that they note that, in XHTML5,

- `<!DOCTYPE>` is mandatory but there is no need to include a DTD
- The `xmlns` attribute in `<html>` is mandatory

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

- `<html>`, `<head>`, `<title>`, and `<body>` are mandatory
- Elements must always be
  - properly nested
  - closed and
  - in lowercase
- Attribute names must always be in lowercase
- Attribute values must always be quoted
- Attribute minimisation is forbidden, i.e. use `disabled="disabled"` rather than `disabled`.

[2] The term HTML5 is retained to indicate that this document refers to post-2011 HTML; in practice, HTML is now a rolling release; so there will be no HTML6.

1.2 Thanks

Thanks to the following for their feedback on this and/or earlier versions.

Alice Kaerast

2 Security

The main sources of security breaches are:

- not validating user input;
- cross-site scripting (XSS);
- SQL injection: in particular, do not allow arbitrary `<img>` attributes or URLs or the insertion of a `<base>` element by a script;
- cross-site request forgery (CSRF): check `origin` headers on all requests;
- clickjacking, often by the insertion of an `<iframe>` element: include checks that the `window` object and the `top` attribute share the same value.

Other sources include:

- visible spoofs based on the similarity of characters in the Unicode set; these can be facilitated by scripts that change CSS font values;
- syntax spoofs;
- non-visible spoofs which need to be addressed by user-agent designers and script writers.

3 The DOCTYPE declaration[3]

```
<!DOCTYPE html>
```

triggers HTML standards mode; it should have no characters before it, in particular no white space; otherwise, some browsers will ignore it. Note also that unnecessary white space between elements in an HTML document can affect the construction of the DOM (Appendix C) and thus the relationship of the nodes which may in turn affect some Javascript scripts (Flanagan, 2011).

All other HTML declarations are obsolete; XHTML declarations are permitted.

In the absence of this declaration, the page will be rendered in HTML quirks mode.

4 The root element

4.1 html

Contains only the `<head>` and `<body>` elements and normally has the `lang=" "` attribute.[4]

Note: the `manifest=" "` attribute giving the location of the application cache manifest is deprecated and should no longer be used; see `rel="manifest"` in <link>.

Figure 1 sets out and attempts to represent the key relationships, if any, between most of the elements in sections 4 to 14.

Where an element supports an ARIA `role=" "` attribute, this is indicated; in the absence of any indication, the ARIA attribute `role="presentation"` is assumed.

[3] Table 1 on page 9 offers a simple template for an HTML page. For an evolving template, go to the HTML5 Boilerplate.

[4] For the full list of language tags, see http://www.iana.org/assignments/language-subtag-registry

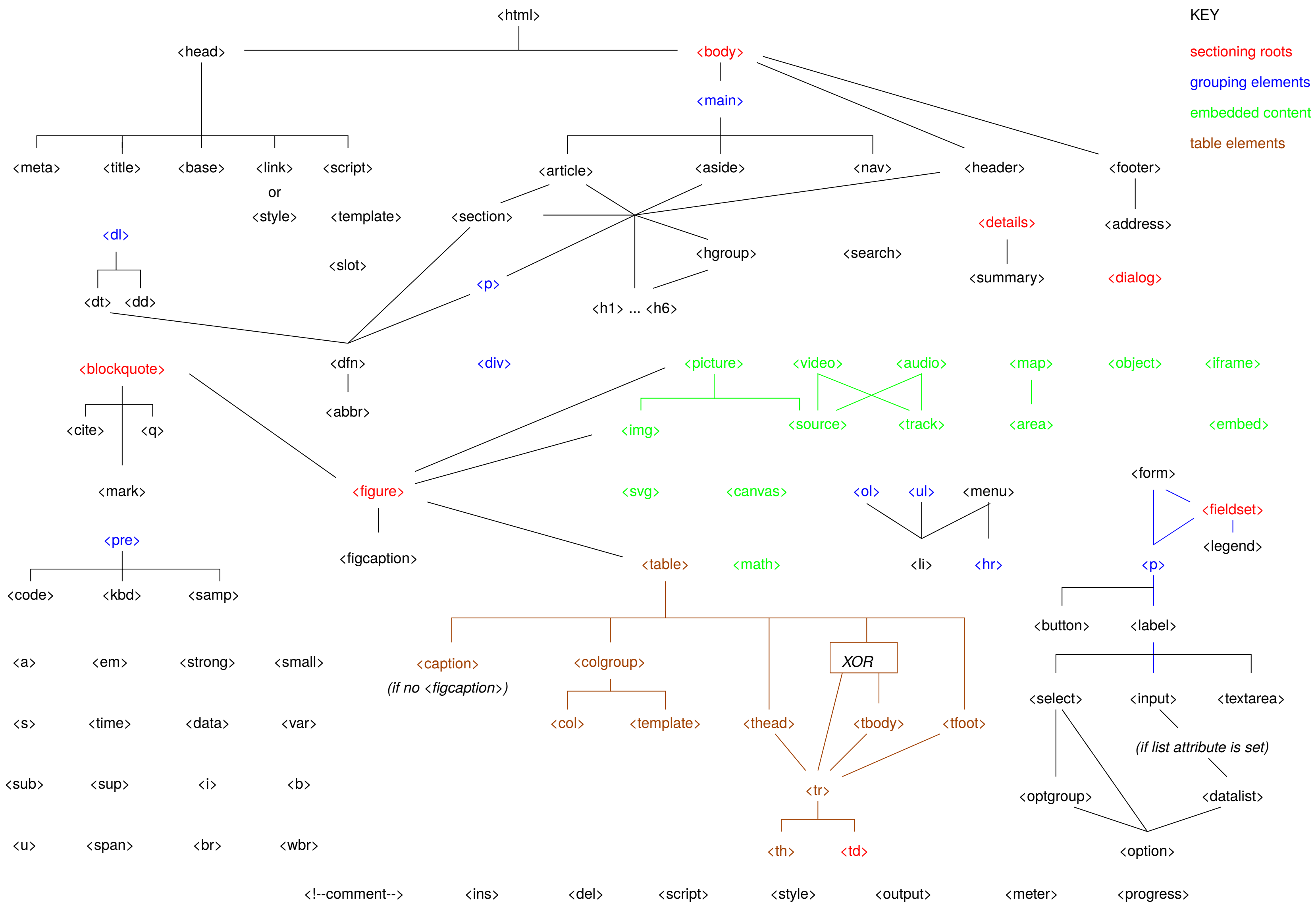


Figure 1: HTML elements

5 The document metadata

5.1 head

This must be the first element in the `<html>` element, it must contain one `<title>` element and may contain no more than one `<base>` element along with any of the other metadata elements. The property `aria-hidden` is set to `true`.

5.2 title

A required element in the `<head>` element and in most child documents; there should only be one `<title>` element per document and it should describe the contents of the document unambiguously, even if the headings are ambiguous, for example, because they are humorous or assume prior knowledge. Most search engines will use the content of the `<title>` element as the basis for searches and most browsers will display it in the frame of the browser.

5.3 base

There must be only one `<base>` element in a document in the `<head>` element; it should contain an `href` or a `target` attribute or both; the `href` attribute must define the document's base URL; the `target` attribute defines the browsing context for the purpose of resolving hyperlinks. It must come before any other elements containing an `href` attribute. It has no end tag.

5.4 link

The `<link>` element defines external resources available to the document; it MUST have

- either a `rel` or an `itemprop` attribute and
- an `href=" "` attribute which defines the source of the external material; for example,

  ```
  <link rel="stylesheet" href="style.css"/>
  ```

points to the styling rules for the page and must be read.

If it has a `rel` attribute, it can only be a child of a `<head>` element and the `rel` attribute may be one of[5]

- `rel="alternate"`
  1. `rel="alternate"` links to an alternate representation of the current document; in this case, it may take the attribute `title=" "` indicating, for example, a different presentation of the material or a different language.
  2. where `rel="alternate"` is combined with `type="application/rss+xml"` or `type="application/atom+xml"` it links to a syndication feed; for example,

     ```
     <link rel="alternate"
         type="application/atom+xml"
         title="My Weblog feed"
         href="/feed/" />
     ```

     points to something external to the page which can be read; the `href` attribute is specified relative to the URL in the `<base>` element.[6]
  3. `rel="alternate stylesheet"` modifies the link created by `rel="stylesheet"`; in this case, it takes the mandatory attribute `title=" "`.
- `rel="canonical"` the preferred URL for the document

[5] The `rel="external"`, `rel="nofollow"`, `rel="norefferer"` and `rel="start"` attributes mentioned by Pilgrim (2010, p. 38f) are not accepted by Hickson (2013) in this context; only `rel="nofollow"`and `rel="norefferer"` are accepted in the context of the `<a>` and `< area>` elements.

[6] This applies to the first `<link>`, `<a>` or `< area>` element in the document.

Table 1: A simple HTML template

```
<!DOCTYPE html>
<html lang="en">

    <head>
       <meta charset="UTF-8"/>
       <meta name=viewport content="width=device-width, initial-scale=1">
       <title>Page title</title>
       <base href="">
       <link rel="stylesheet" type="text/css" href="default.css"/>
    </head>
    <body>
    ...
    </body>

</html>[a]
```

[a] HTML5 Boilerplate offers a range of HTML templates which help in configuring servers and contain useful example scripts but they also contain some deprecated elements and some non-conforming attributes.

- `rel="author"` links to a document describing the current document's author or her/his email address
- `rel="dns-prefetch"` preemptively perform DNS resolution for the target resource's origin
- `rel="help"` creates a hyperlink to context sensitive help for the page as a whole
- `rel="icon"` creates a hyperlink to an icon on an external resource which represents the current document
- `rel="license"` creates a hyperlink to a licence document; the author must indicate whether it is applicable to the whole document or to an element within the document; the legacy value `copyright` is treated in the same way
- `rel="manifest"` imports or links to an application manifest
- `rel="modulepreload"` is a specialised alternative to `rel="preload"`; preemptively fetch the module script and store it in the document's module map for later evaluation
- `rel="next"` creates a hyperlink to the next document in the series to which the current document belongs
- `rel="pingback"` creates a hyperlink to the pingback server that handles pingbacks to the current document
- `rel="preconnect"` preemptively connect to the target resource‘s origin
- `rel="prefetch"` preemptively fetch and cache the target resource
- `rel="preload"` preemptively fetch and cache the target resource for current navigation according to the potential destination given by the `as=" "` attribute

- `rel="prev"` creates a hyperlink to the previous document in the series to which the current document belongs; the legacy value `previous` is treated in the same way
- `rel="search"` creates a hyperlink to a resource which can be used to search the current document
- `rel="stylesheet"` links to a stylesheet

If `rel="icon"`, the attribute `sizes=" "` contains one or more space separated sizes for the available icons where `"any"` indicates a scalable icon, for example, SVG.

If `rel="modulepreload"` or `rel="preload"` the attribute `as=" "` defines the potential destination for the preload request.

If `rel="modulepreload"`, `rel="preload"` or `rel="stylesheet"` the `integrity=" "` attribute contains integrity metadata.

If `rel="preload"` and it has the attribute `as=" "` in the "image" state, it may take the following attributes:

- `imagesrcset=" "` specifies a `srcset=" "` that is, a comma separated list of URLs each optionally followed by one of:

  - a width descriptor of the form *n*`w`
  - a height descriptor of the form *n*`h`
  - a density descriptor of the form *n*`x`

  where the unit is CSS pixels; a density descriptor for the widest image is preferred for legacy mobile 'phones but a width descriptor should be used where image size is to depend on the size of viewport. The purpose of this attribute is to allow user agents to select the image that best suits a device; therefore no two images should have the same dimensions.
- `imagesizes=" "` specifies a value for `sizes=" "` that, is, a comma separated list of the available image sizes, where any of the images in the `srcset=" "` element has a width descriptor; each size must include a width descriptor which may be preceded by a CSS declaration in parentheses which acts as a condition for the use of a particular size.

If `rel="stylesheet"` it may take the attribute `blocking=" "`

If it has the `itemprop` attribute, it may also be used within the `<body>` element within the constraints of Microdata (section 16 on page 52).

The other attributes may be

- `crossorigin` allows cross origin access; it may have one of the values:

  - `anonymous`
  - `use-credentials`

  if used, it is recommended that strict access precautions are taken to prevent port scans
- `media=" "` specifies the media using one of `all`, `print` or `screen` in parentheses; the default value is `all`
- `hreflang=" "` defines the language of the external material
- `role="link"` an ARIA only attribute for use where the element creates a hyperlink and the content of the element does not make this clear
- `type=" "` specifies the MIME type of the resource; this is purely advisory and browsers are expected to determine the MIME type of a resource themselves and not to load any MIME type that they do not support.
- `referrerpolicy=" "` for any fetches initiated by the element
- `fetchpriority=" "`
- `title=" "` optional; if it is absent, the element does not inherit its value from its parent but, if `link rel="alternate stylesheet"`, it defines the alternative stylesheet sets

A `<link>` element may have the `autofocus` attribute; see autofocus on page 39.

It has no end tag.

5.5 meta

> The meta element represents various kinds of metadata that cannot be expressed using the title, base, link, style, and script elements (Hickson, 2011, p. 123).

It, or they, should come before all the other elements in the `<head>` element (table 1) and

- `charset=" "` should take the attribute `UTF-8` for newly created documents — it should always be specified because leaving it out creates security vulnerabilities —

  while the remainder may contain one each of the following attributes:

- `http-equiv=" "` use of this attribute is discouraged as there are alternative ways of achieving what it does; it may take the attributes:
  - `content-language` n.b. non-conforming; use the `lang=" "` attribute in the `<html>` element instead
  - `content-type` n.b. use the `charset=" "` attribute instead
  - `default-style` the preferred stylesheet set, in which case the `content=" "` attribute should hold the preferred stylesheet set
  - `refresh` refreshes the page every *n* seconds where *n* is specified in `content="`*n*`[; URL= ]"`, which optionally specifies an alternative page to load
  - `set-cookie` n.b. non-conforming; use HTTP headers instead
  - `x-ua-compatible` takes the attribute `content="IE=edge"`
  - `content-security-policy` specifies in the attribute `content=" "` the Content Security Policy
- `itemprop=" "` see Microdata on page 52
- `media=" "`
- `name=" "` which may have as its values:
  - `application-name` in which case `content=" "` should hold the name of the Web application that created the website
  - `author` in which case `content=" "` should hold one author's name
  - `description` in which case `content=" "` should hold a description of the page suitable for using in a directory of pages
  - `generator` in which case `content=" "` should hold the name of the software that created the page; it should not be used where the page was created in a text editor
  - `keywords` in which case `content=" "` should contain a comma separated list of key words; n.b. many search engines ignore this attribute because it has historically been used to spam search engine results
  - `referrer` in which case `content=" "` contains a value for referrer policy or a legacy value where
    - ∗ 'never' is evaluated as `no-referrer`,
    - ∗ 'default' as `no-referrer-when-downgrade`,
    - ∗ 'always' as `unsafe-url` and
    - ∗ 'origin-when-crossorigin' as `origin-when-cross-origin`
  - `theme-color` in which case `content=" "` defines the CSS colour which may be used; it may take the attribute `media=" "` saying to which media the metadata applies
  - `color-scheme` in which case `content=" "` defines the colour scheme which may be used.

TOGETHER WITH

- `content=" "` which holds the data relating to each of the three attributes above, if present.

The `<meta name=" " content=" ">` mechanism is intended to enable authors to register extensions to the predefined set of metadata names. In particular, Google recommends:

```
        <meta name=viewport content="width=
        device-width, initial-scale=1">
```

be used in mobile friendly websites to overcome a bug in iOS. It has no end tag.

5.6 style

In the `<head>` element it is an alternative to a `<link rel="stylesheet" href=" ">` statement (page 8), for example,

```
<style> body{color: black; background: white;}
    em{font-style: normal; color: red;}
</style>
```

It may take the following attributes:

- `media=" "` specifies the media using a CSS descriptor in parentheses (which may be `all`, `print` or `screen`); the default value is `all`
- `title=" "` defines alternative stylesheet sets; in its absence the element does not inherit its parent's `title` attribute
- `type=" "` the default value is `text/css`

If it appears outside the `<head>` element, it may not appear in an `<a>` or `<ins>` element (see ins and del on page 25), it must take the `scoped` attribute, a Boolean attribute which means that the style is only applied to the parent and child elements, not to the whole document, and it must appear before any content in its parent element.

`<style>` elements with the `scoped` attribute may contain CSS `@global` rules but CSS `@page` rules will be ignored.

However, the use of `<style></style>` statements is deprecated in favour of `<link rel="stylesheet" href=" ">` statements other than for rapid prototyping prior to removal to a stylesheet.

The property `aria-hidden` will be set to `true`.

6 Sectioning elements

In general elements should not cross typographical paragraph boundaries; where this is not clear use explicit `<p></p>` elements to define the boundaries typographically. Tags should also be nested and should not overlap each other.

6.1 body

There is only one `<body>` in a document; it is the second element in an `<html>` element.

Five of the HTML semantic elements play a key role in the layout of the `<body>` element (see figure 2 on page 14).

The `<body>` element should only include document or application elements and the ARIA attribute `role=" "` may take one of the values:

- `document` where it contains a focusable child element that assistive technology users may want to browse in a reading mode
- `application` where it contains one or more focusable elements requiring user input, such as keyboard or gesture events.

The `<body>` element may take any of the attributes: `onafterprint`, `onbeforeprint`, `onbeforeunload`, `onhashchange`, `onlanguagechange`, `onmessage`, `onmessaggerror`, `onoffline`, `ononline`, `onpagehide`, `onpageshow`, `onpopstate`, `onrejectionhandled`, `onstorage`, `onhandledrejection` and `onunload` which trigger the relevant event handlers. However, onblur, onerror, onfocus, onload, onresize, and onscroll are handled by the Window event handler unless addEventListener() is used to attach an event listener to the <body> element.[7]

6.1.1 A note on sectioning roots

Seven elements are considered sectioning roots, that is, any child sections and headings within these elements do not contribute to the outlines of their ancestors:

- body
- blockquote
- details
- dialog
- fieldset
- figure
- td

6.2 article

The <article> element is normally a self-contained composition which could easily be detached from its context. A nested <article> might include comments on the article but should always relate directly to the content of the parent <article>.

It normally takes the ARIA attribute role="article" but a scrollable list of articles where scrolling may cause articles to be added to or removed from either end of the list takes the ARIA attribute role="feed".

6.3 section

The <section> element contains material relating to a particular theme within a document or application; it may be used within an <article> element or for the sections of a <body> where <article> is not appropriate but not for

- material which might be independently syndicated where <article> is more appropriate
- very short pieces of content where < div> is more appropriate.

One criterion for distinguishing <section> and <div> is that a <section> element might appear in a table of contents but a <div> element would not.

A <section> element within a <body> element is an appropriate container for an <hgroup> whereas, within an <article> element, it might share the same <h1--h6> elements.

A <section> element cannot appear within a < kbd> element.

It normally takes the ARIA attribute role="region" when it also takes aria-labelledby=" " if a visible label is present or aria-label=" " if not.

6.4 nav

The <nav> element contains navigation links to other pages or to other parts of the same page (see the examples of the use of href=" " in <a> on page 20); it may also contain explanatory text. A <nav> element is an appropriate container for an <hgroup> and/or <h1--h6> elements but may not contain a < main> element. A <nav> element containing a group of minor links may also appear within another sectioning element, for example, the <footer> element or the <nav> element may be omitted completely in favour of using the <footer> element only for such links.

It normally takes the ARIA attribute role="navigation".

[7] See *Some notes on Javascript*.

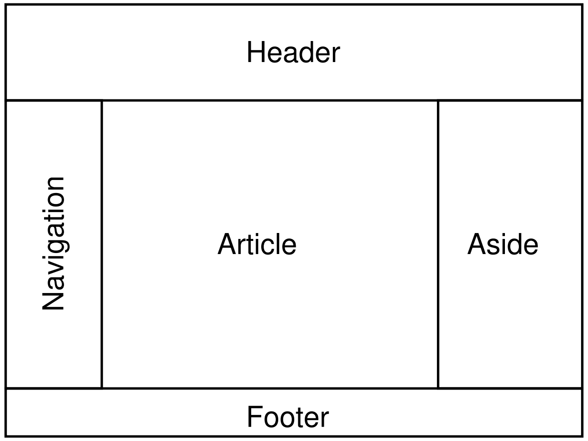


Figure 2: The relationship of five of the elements in the `<body>` element (thanks to David Fisher (2010))

6.5 aside

An `<aside>` element contains content that is tangentially related to what is around it. It may contain `<nav>` elements related to this tangential content but may not contain a `<main>` element. It should not be used for parenthetical comments or for supplementing the main content.

It normally takes the ARIA attributes `role="complementary"` and `aria-labelledby=" "` if a visible label is present or `aria-label=" "` if not.

6.6 `h1--h6`

The `<h1>`–`<h6>` elements represent a hierarchy of headings within any type of section in a document. While two or more `<h1>`–`<h6>` elements of the same rank within a sectioning element implicitly create new sections, authors are encouraged to wrap these sections in explicit sectioning elements.[8] Note that heading elements within a sectioning root (section 6.1.1 on the previous page) do not interact with the outlines of their ancestors.

An `<h1>--<h5>` element in a child element is equivalent to an `<h2>--<h6>` element in its parent element; an `<h1>--<h4>` element in a child element is equivalent to an `<h3>--<h6>` element in its grandparent element, and so on.

If they have no `hgroup` ancestor, the property `aria-level` will be set to the element's outline depth.

[8] Note that using an `<h1>--<h6>` element in a `<li>` element implicitly splits the list element into two sections.

6.7 hgroup

The <hgroup> element may contain <h1>–<h6> elements which allows you to create multilevel headings within an element which do not take effect outside that element, for example,

```
<header>

    <hgroup>
        <h1>My Weblog</h1>
        <h2>A lot of effort went into making this effortless.</h2>
    </hgroup> ...

</header>
```

It may also be used where there is more than one level of <h1>--<h6> elements within a parent element in order to avoid unexpected interactions with <h1>--<h6> elements in other elements. Note that heading elements within a sectioning root (section 6.1.1 on page 13) do not interact with the outlines of their ancestors.

It may take the ARIA attribute role="heading" in which case the property aria-level is set to its outline depth.[9]

6.8 header

The <header> element is intended to provide a heading for a sectioning element; it cannot be a free-standing section itself. It normally contains an <h1> or <hgroup> element and may include a table of contents, search form, logo or navigational aids. A header can come in any type of section including an <h1>–<h6> element but it may not contain another <header> element, a <footer> element or a < main> element.

It takes the ARIA attribute role="banner" when it is a child of the <body> element.

6.9 footer

A <footer> element is intended to provide content information relating to the nearest sectioning element, such as author, copyright information and links; it represents a section within another sectioning element rather than being a free-standing section itself. A document may have multiple <footer> elements related to the <body> and to individual sectioning elements within the document but a <footer> element may not contain another <header> element, a <footer> element or a < main> element.

The <footer> element is not intended to hold footnotes for which the title=" " attribute of the relevant element is usually most appropriate for short footnotes and the <a href=" "> construct for longer ones. Alternatively, the <caption> or <figcaption> elements can be used for the footnotes included in cells at the foot of a table.

It takes the ARIA attribute role="contentinfo" when it is a child of the <body> element.

[9] This, however, in not recommended in the current *Accessible Rich Internet Applications (WAI-ARIA) 1.1*.

6.10 address

An `<address>` element contains contact information relating to the nearest section statement. It must not contain any other type of information and it would normally appear within a `<footer>` element, for example,

```
<footer>
    <address>For more details, contact <a href="mailto:js@example.com">John Smith</a>.</address>
</footer>
```

but may appear in a `<body>` or `<article>` element which has no `<footer>` element.

An `<address>` element may contain no other sectioning elements or sectioning content and no `<header>`, `<footer>` or `<address>` elements.

It normally takes the ARIA attribute `role="contentinfo"`. Where there are multiple elements with the `contentinfo` attribute, use `aria-labelledby=" "` if a visible label is present or `aria-label=" "` if not to distinguish them.

---

7 Grouping elements

7.1 p

The `<p>` element represents a paragraph. It should normally be used to represent conversations but should not be used instead of a more appropriate element. It cannot contain list elements such as `<ol>` and `<ul>`.

7.2 hr

The `<hr>` element provides a separator between two paragraphs indicating a thematic change between the content of the paragraphs which cannot be expressed in any other way, for example, by using a `<section>` or `h1--h6` elements. It has no end tag.

It takes the ARIA attribute `role="separator"`. Where there are multiple elements with the `separator` attribute, use `aria-labelledby=" "` if a visible label is present or `aria-label=" "` if not to distinguish them.

7.3 pre

The `<pre>` element allows the introduction of pre-formatted text into a document; it would typically be used with the `< code>`, `< kbd>` or `< samp>` elements but could, for example, be used to display the contents of an email or specifically formatted poetry.

The `<pre>` element can also be used to include HTML code in a page by substituting `&lt;` for `<` and `&gt;` for `>`; for example, `<pre>&lt;meta charset="UTF-8"&gt;</pre>` will display as

```
<meta charset="UTF-8">.
```

However, there is no consistency in the way browsers render this element and the impact of any pre-formatting will be lost on users of speech synthesisers, Braille displays, etc.

7.4 blockquote

The `<blockquote>` element contains content from another source and may have a `cite=" "` attribute containing the URL of the source. The author should be given in a separate element, for example, <p>, or a <figcaption> element where the <blockquote> element is a child of a <figure> element. Use the <cite> element for the title of sources which cannot be expressed as URLs.

<blockquote> should not be used to represent conversations; <p> should be used instead.

7.5 ol

The <ol> element contains an intentionally ordered list. It should not be used for lists where the order is immaterial. It normally has at least one <li> element and may have the attributes:

- reversed a Boolean attribute; when set, the list is output in descending order
- start= takes an integer as the start value of the list

  If the first <li> element contains a value= attribute that can be parsed, this will be used in preference to anything else and each subsequent <li> element will be numbered one more, or one less if the reversed attribute has been set, unless it too contains a value= attribute. Otherwise, the start= attribute value, if any, will be used.
- type=" " may take the value:
  - 1 decimal numbers; the default
  - a lowercase Latin alphabet
  - A uppercase Latin alphabet
  - i lowercase Roman numerals
  - I uppercase Roman numerals

  but is not needed where the corresponding values have been set as attributes of the list-style-type property in a CSS style sheet (note that the list-style-type property offers considerably more options including those created using an @counter-style rule).

The child elements are always <li> or scripting elements.

It normally takes the ARIA attribute role="list". Where there are multiple elements with the list attribute, use aria-labelledby=" " if a visible label is present or aria-label=" " if not to distinguish them.

7.6 ul

The <ul> element contains a list where the order is immaterial. The child elements are always <li> or scripting elements. The default list-style-type is a disc but alternatives may be displayed by using the list-style-type property in a CSS style sheet.

It normally takes the ARIA attribute role="list". Where there are multiple elements with the list attribute, use aria-labelledby=" " if a visible label is present or aria-label=" " if not to distinguish them.

7.7 menu

The <menu> element is a semantic alternative to the <ul> element containing a list of commands within <li> elements displayed as a toolbar.

It takes the ARIA attribute role="toolbar". Where there are multiple elements with role="toolbar", use aria-labelledby=" " if a visible label is present or aria-label=" " if not to distinguish them.

7.8 li

The <li> element contains a list item within a <menu> element whose type="toolbar", an <ol> or a <ul> element. When it is within an <ol> element, it may have the attribute value= where the value is an integer which gives the element an ordinal value.

A <li> element within a <menu> element respects the :enabled and :disabled pseudo-classes of CSS.

It may, but does not need to, take the ARIA attribute `role="listitem"` when its parent is an `<ol>` or `<ul>` element.

7.9 dl, dt, dd

The `<dl>` element contains a datalist consisting of associated `<dt>` (term) and `<dd>` (definition) elements. There must be at least one `<dt>` element before one or more `<dd>` elements but associations may be one to one, one to many or many to one allowing multiple definitions of a term and multiple terms for one definition. To indicate that the term in the `<dt>` element is being formally defined in the `<dd>` element, enclose it in a `<dfn>` element within the `<dt>` element:

```
<dl>

    <dt><dfn>Home</dfn>, n.</dt>
    <dd>The user’s login directory.</dd>

</dl>
```

A `<dt>` element may contain no sectioning or heading elements or a `<header>` or `<footer>` element but may contain `< div>` elements used to group `<dt>` and `<dd>` elements.

The `<dd>` element takes the ARIA attribute `role="definition"`. Use `aria-labelledby="  "` to reference the term.

7.10 figure

The `<figure>` element contains content which can be displayed separately from the content of which it is a part, such as illustrations, diagrams, photos or code listings (*cf.* float in LaTeX). It may contain a `<figcaption>` element as either the first or the last child element providing a caption for the content.

A `<figure>` element containing a `<blockquote>` and a `<figcaption>` element may he used to hold a direct quotation followed by its citation.

Where multiple images are displayed use the `title="  "` attribute to provide individual captions and the `<figcaption>` element for the main caption.

Use the `<aside>` element for content which is tangential to or does not relate to the flow of the main content of the element.

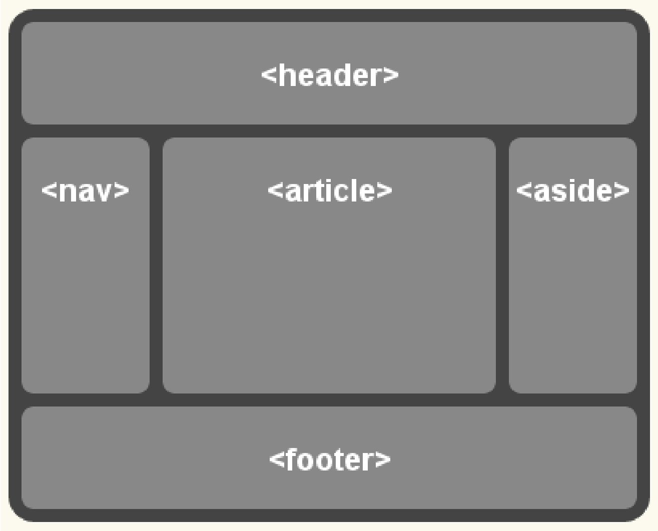
The `<figure>` element may take the ARIA attribute `role="figure"` and must take the ARIA attribute `aria-label="  "` in the absence of a `<figcaption>` element to give it a title for the benefit of screen-reader users.

The `<figcaption>` element should take the ARIA attribute `aria-labelledby` on the parent `figure`. However, if a `caption` contains content that serves as both a name and description for its parent, authors MAY instead set `aria-labelledby` to reference an element within the `caption` that contains a concise name, and set `aria-describedby` to reference an element within the `caption` that contains the descriptive content.

7.11 main

The `<main>` element contains the dominant content of a document and is normally the child of a `<body>` element but can only be a child of an `<html>`, `<body>`, `< div>` or `<form>` element; there can be only one `<main>` element in a document.

It is particularly suitable as a flex container for the semantic elements (figure 2 on page 14).



Using the HTML structure:

```
<body>

    <header> ... </header>
    <main>
        <article> ... </article>
        <nav> ... </nav>
        <aside> ... </aside>
    </main>
    <footer> ... </footer>

<body>
```

and the CSS declarations:

```
main {display: flex;}
main > article {order: 2; ...}
main > nav {order: 1; ... }
main > aside {order: 3; ... }
```

ensures that the <article>, <nav> and <aside> elements will be of equal height and that a screenreader will read the <article> element before the <nav> element while a browser will display the <nav> element first (Source: CSS Flexible Box Layout).

In takes the ARIA attribute `role="main"`, which offers a non-obtrusive alternative to 'skip to main content' links, when it is a child of the <body> element. If the content of the element changes, use `aria-labelledby=" "` if a visible label is present or `aria-label=" "` if not.

7.12 search

The <search> element contains a set of form controls or other content related to performing a search or filtering operation.

7.13 div

The <div> element is the element of last resort and should only be used to distinguish content where there is no alternative element available or as the child of a <dl> element (see dl, dt, dd).

It can take the attributes:

- `class=" "`
- `lang=" "`
- `title=" "`

in order to distinguish the content.

It is normally rendered as a block box and cannot be placed within a < span> element which is an inline box.

8 Text level semantics

8.1 a

The `<a>` element creates a placeholder for a hyperlink which is activated by an `href=" "` attribute which must be present for the element to have any of the attributes:

- `target=" "` which activates a browsing context (see Other notes on attributes)
- `download` downloads a resource from a hyperlink; it may have as a value the filename to be given to the downloaded file; this must be chosen carefully to reflect the different conventions used by different filesystems
- `ping=" "` contains the URLs of resources that are interested in knowing if the user follows the hyperlink; authors are encouraged to use this rather than HTTP redirects or Javascript because it can be disabled by the user
- `rel=" "` make take the values:
    - `alternate` links to an alternate representation of the current document; if `hreflang=" "` is different from the root language of the document, it means that the linked content is a translation; if the `type=" "` attribute is not the same as the `type=" "` attribute of the parent element, the linked content will be an alternative format of the the current document.[10]
    - `author` links to a document describing the current section's author or her/his email address
    - `bookmark` creates a hyperlink to the nearest ancestor section
    - `external` indicates that the referenced document is on a different site
    - `help` creates a hyperlink to context sensitive help for the parent element
    - `license` creates a hyperlink to a licence document; the author must make clear the element to which it applies
    - `next` creates a hyperlink to the next document in the series to which the current document belongs
    - `nofollow` the link is not endorsed by the author/publisher of the current document
    - `noopener` the browsing context created by following this link is disowned
    - `noreferrer` requires the user agent not to send an HTTP Referer [*sic*] header with the link, preventing display of the URL from which this URL was accessed (that is, it prevents use of the Javascript `d.referrer` document property)
    - `opener` creates an auxiliary browsing context, i.e. one that has `"_blank"` as a `target` attribute value
    - `prev` creates a hyperlink to the previous document in the series to which the current document belongs; the legacy value `previous` is treated in the same way
    - `search` creates a hyperlink to a resource which can be used to search the current document
    - `tag` creates a hyperlink to an address which gives a tag describing the current document
- `hreflang=" "` defines the language of the linked content
- `type=" "` gives the MIME type of the linked content
- `referrerpolicy=" "` for any fetches initiated by the element
- `itemprop=" "` see Microdata on page 52

`href=" "` may refer to an external or an internal resource using:

[10] See also the footnote on page 8.

- href="/" for the page pointed at by the <base> element
- href="/<path>/" for the page pointed at by the value of <path> relative to the value of the <base> element
- href="#<value>" for the element on the same page whose id="<value>"
- href="?<value>" for the action whose value is <value>

An <a> element may have the autofocus attribute; see autofocus on page 39.

An <a> element may have child content elements as long as they have no interactive content so that a whole block may become a link.

It may not contain a <style> element.

In the presence of a valid hyperlink, it takes the ARIA attribute role="link" if the content of the element does not make it clear what its function is.

In the absence of a valid href=" " attribute, it must not have any other attributes but it may contain another element as long as that element contains no interactive content.

See also section 14.4 on page 48.

8.2 em

The <em> element stresses the content it encloses; it should not be used to indicate a different mood or to indicate importance for which <i> and <strong> respectively are more appropriate.

8.3 strong

The <strong> element marks the content it encloses as important, serious or urgent without changing its meaning.

8.4 small

The <small> element indicates side content such as disclaimers, caveats, legal restrictions, or copyrights which are not part of the main content. It might fall within an <aside> element. It should not be used for multiple paragraphs or headings or for styling extended sections of text.

8.5 s

The <s> element indicates content that has been superseded by new content, for example, a reduction in a price. It should not be used for edited text for which <del> should be used.

8.6 cite

The <cite> element contains the title only of a work; that is, it should only enclose the title of the work, not the authors name or a whole citation.

8.7 q

The <q> element contains a direct quotation from a source which may be specified as a URL in a cite=" " attribute. Use the <cite> element for the title of sources which cannot be expressed as URLs. It should not be used to provide quotation marks around content which is not a direct quotation.

It is an alternative to typographical quotation marks which should not be used around a <q> element which will provide the necessary quotation marks. These will be rendered in the language of the parent element making the <q> element useful in multi-lingual matter.

<q> elements may be nested.

8.8 dfn

A `<dfn>` element holds a term which is being defined in a `<p>`, `<dl>` or `<section>` element. It may have a `title=" "` attribute giving the precise term being defined but, if the `<dfn>` element contains an `<abbr>` element, that must contain the `title=" "` attribute instead; for example,

```
<dfn><abbr title="Garage Door Opener">GDO</abbr></dfn>
```

It takes the ARIA attribute `role="definition"`.

8.9 abbr

The `<abbr>` element contains an abbreviation and may contain its expansion as a `title=" "` attribute. It is not intended to hold all abbreviations but those which might need to be clarified, explained or differently styled.

8.10 ruby, rt and rp

The `<ruby>`, `<rt>` and `<rp>` elements are used to annotate certain east Asian languages; see Hickson (2013, 4.6.21–23) and Internationalization techniques: Authoring HTML and CSS.

8.11 data

The `<data>` element contains human readable data along with a `value=" "` attribute containing the same data in machine readable form which must be present. It is in part intended to provide machine readable data for use by Microdata (section 16 on page 52).

8.12 time

The `<time>` element holds a human readable date and/or time with an optional time zone; it may have the attribute `datatime=" "` whose value is a machine-readable timestamp consisting of, *inter alia*,

- parts of the form `yyyy-mm-dd` which may be extended with successive parts of `[T]hh:mm:ss[Z]±hh:mm` where `T` optionally introduces the time and `Z` for no offset from UTC or ± the offset from UTC or
- a week string, of the form `yyyy-W`*nn* or
- a duration string of the form *n*`h` *n*`m` *n*`s`

for example,

```
<time datetime="2015-08-10T18:00+01">
August 10, 2015 at 7pm BST</time>
```

8.13 code

The `<code>` element contains computer code; its language can be specified with the `class=" "` attribute, where the attribute begins `language-`.

8.14 var

The `<var>` element holds a variable; it is intended for use in prose contexts; MathML should be used for mathematical expressions.

8.15 samp

The `<samp>` element represents the output from a computer program or system.

8.16 kbd

The `<kbd>` element contains keyboard input which may contain further `<kbd>` elements representing keys to be pressed; when inside a `<samp>` element, it represents input echoed by the system; when it encloses a `<samp>` element, the `<samp>` element represents menu choices. It may not contain a `<section>` element.

8.17 sub and sup

The `<sub>` and `<sup>` elements contain content in sub- or superscript position; they are intended for use in prose contexts and may be used with the `<var>` element for mathematical sub- and superscripts in such contexts. They should not be used for presentational purposes.

8.18 i

The `<i>` element is used to indicate an alternative mood or voice or to differentiate content within the main content, for example, words in a foreign language or transliterations; it may or may not be rendered as *italic* and should not be used where the `<em>` or the `<dfn>` elements are more appropriate; it may take the attributes:

- `class=" "` to indicate the grounds for the differentiation
- `lang=" "` to indicate the language of the content.[11]

Note that using the `<i>` element with the `lang=" "` attribute to indicate words in a foreign language causes any `<q>` elements within the element to display language appropriate quotation marks.

8.19 b

The `<b>` element is used to differentiate content which is not different, whether in importance or mood, from the main content, for example, keywords or the lede, or leading words in a paragraph; it may take the attribute:

- `class=" "` to indicate the grounds for the differentiation.

It may or may not be rendered as **bold** and it should not be used where `<em>`, `<h1--h6>`, `<mark>` or `<strong>` are appropriate.

8.20 u

The `<u>` element contains text that is different from the normal flow for a reason not covered by an `<em>`, `<cite>`, `<i>`, `<b>`, `<mark>` or `<ruby>` element, for example, a misspelling. However, because of the potential confusion with a hyperlink, care should be taken with its use.

8.21 mark

The `<mark>` element contains content, normally from a third party source, that has been highlighted by the author to draw attention to it, sometimes by restyling it, while retaining the original styling for the rest of the content.

8.22 bdi and bdo

The `<bdi>` and `<bdo>` elements contain bidirectional content. The `<bdi>` element is useful for embedding content whose directionality is unknown; it takes the `dir=" "` attribute which defaults to `auto`. The `<bdo>` element contains content whose directionality is known; it must have the `dir=" "` attribute set to either `ltr` or `rtl`; see Hickson (2013, 4.6.24–25) and Internationalization techniques: Authoring HTML and CSS.

8.23 span

The `<span>` element is considered as an inline box: it normally takes one or more of the attributes:

- `class=" "`
- `dir=" "`
- `lang=" "`

It might be used, for example, so that a short piece of text might be coloured differently or to associate Microdata name/value pairs within a single element.

8.24 br

The `<br>` element represents a line break. It should only be used where other elements or CSS rules would not automatically create a line break anyway, for example, verse or addresses. It has no end tag.

[11] For the full list of language tags, see http://www.iana.org/assignments/language-subtag-registry

8.25 wbr

The `<wbr>` element represents an optional line break where the normal algorithms for line breaks would not work. It has no end tag.

9 Editing elements

9.1 ins and del

The `<ins>` and `<del>` elements contain material that has been added or deleted since the original document was created; they may take the attributes:

- `cite=" "` containing a URL pointing to a document explaining the change
- `datetime=" "` to indicate when the insertion or deletion took place

The content they enclose should not cross paragraph boundaries. Where necessary, explicit paragraph boundaries should be defined using the `<p>` element. The contents of `<ol>` or `<ul>` elements or of tables cannot be inserted or deleted but the content of a `<li>` element, a table row or a table column can be modified; a `<del>` element should contain all the content of the `<li>` element, table row or table column to be deleted and all the content of a new `<li>` element, table row or table column inserted after the `<del>` element should be contained within an `<ins>` element.

`<ins>` may not contain a `<style>` element.

10 Embedded content

10.1 picture

The `<picture>` element contains, normally, multiple `<source>` elements from which the browser can choose the image source which most closely matches the screen density, viewport and image formats supported by the display followed by an `<img>` element.

10.2 source

The `<source>` element is a child of a `<picture>` or media element intended to specify one of a number of alternative sources for the element to enable the browser to determine the most suitable image format or whether it can play a media element and only download it if it can.

If it is the child of a `<picture>` element, it may take the attributes:

- `srcset=" "` a comma separated list of URLs each optionally followed by one of:
  - a width descriptor of the form *n*`w`
  - a height descriptor of the form *n*`h`
  - a density descriptor of the form *n*`x`

  where the unit is CSS pixels; a density descriptor for the widest image is preferred for legacy mobile 'phones but a width descriptor should be used where image size is to depend on the size of viewport. The purpose of this attribute is to allow user agents to select the image that best suits a device; therefore no two images should have the same dimensions.
- `sizes=" "` a comma separated list of the available image sizes, where any of the images in the `srcset=" "` element has a width descriptor; each size must include a width descriptor which may be preceded by a CSS declaration in parentheses which acts as a condition for the use of a particular size
- `media=" "` specifies the media using a CSS descriptor in parentheses (which may be `all`, `print` or `screen`); the default value is `all`
- `width=" "` where there is a `srcset=" "` attribute but no `sizes=" "` attribute; this will be used in preference to any similar attribute in an `<img>` element.
- `height=" "` where there is a `srcset=" "` attribute but no `sizes=" "` attribute; this will be used in preference to any similar attribute in an `<img>` element.

OR, if it is a child of a media element, the attribute

- `src=" "` the address of the media resource

AND, if it is either, the attribute

- `type=''` which specifies the MIME type of the
  - images in the `srcset=" "` attribute where the `<source>` element is a child of a `<picture>` element; omitting it may mean the browser does not look for a more suitable `<source>` element
  - media element in the `src=" "` attribute followed after a `;` by `codecs=" "` which contains a comma separated list of codecs to match the value of the `type=''` attribute where the `<source>` element is a child of a media element.

Where it is:

- a child of a `<picture>` element, it must take the `srcset=" "` attribute and may take the `media=" "` or `type=''` attributes or both;
- a sibling of a `<source>` element within a `<picture>` element, it must take the `type=''` attribute and may take the `media=" "` attribute;
- a child of a media element, it must take the `src=" "` and `type=''` attributes.

The `type=''` attribute is useful for preventing browsers from downloading content which they cannot display or play, for example,[12]

```
<video width="320" height="240" controls>

    <source src="pr6.mp4" type='video/mp4; codecs="avc1.42E01E, mp4a.40.2"'>
    <source src="pr6.webm" type='video/webm; codecs="vp8, vorbis"'>
    <source src="pr6.ogv" type='video/ogg; codecs="theora, vorbis"'>

</video>
```

[12] A bug in some iPads means that they only see the first entry; so the MP4 source should always come first.

where

- the *container* can be
  - mp4
  - webm
  - ogg
- *video codec* can be
  - avc1.42E01E
  - vp8
  - theora
- *audio codec* can be
  - mp4a.40.2
  - vorbis

It has no end tag.

Further examples are given in appendix E on page 79.

Additionally, add the following lines to httpd.conf if using an Apache server, because not all browsers interpret the video MIME types correctly:

```
AddType video/ogg .ogv
AddType video/mp4 .mp4
AddType video/webm .webm
```

10.3 img

The <img> element[13] is a container for an image; it must have a src=" " attribute; it also should have an alt=" " attribute providing a textual alternative to the image; this should not be the equivalent of the <figcaption> or <legend> elements or the title=" " attribute (which may be used to provide advisory information or a caption if the image is not in a <figure> element) but something which directly replaces the image. For example, it should say what an image in a <button> element represents. The alt=" " attribute may be empty if the image is

- in a <figure> element which also contains a <figcaption> element and the title=" " attribute is not empty
- adjacent to text which provides the same information
- a logo which conveys no additional information.

In the last two cases, the ARIA attribute role="presentation" should be used; otherwise, the default is role="img". It has no end tag.

Each source image must be a single page but may be animated; the <img> element considers any sibling <source> elements in a <picture> element in deciding which image to load; it may take the following two attributes before the src=" " attribute:

- sizes=" " a comma separated list of the available image sizes, where any of the images in the srcset=" " element has a width descriptor, in which case there must be no width=" " or height=" " attribute; each size must include a width descriptor which may be preceded by a CSS declaration in parentheses which acts as a condition for the use of a particular size
- srcset=" " a comma separated list of URLs each optionally followed by one of:
  - a width descriptor of the form *n*w
  - a height descriptor of the form *n*h
  - a density descriptor of the form *n*x

  where the unit is CSS pixels; a density descriptor for the widest image is preferred for legacy mobile 'phones but a width descriptor should be used where image size is to depend on the size of viewport. The purpose of this attribute is to allow user agents to select the image that best suits a device; therefore no two images should have the same dimensions.

[13] See *Some notes on Javascript* for the use of images in the < canvas> element.

along with the following attributes:

- crossorigin allows cross origin access, for example, to images used in a <canvas> element; it may have one of the values:

  - anonymous
  - use-credentials

  if used, it is recommended that strict access precautions are taken to prevent port scans.
- usemap=" " whose value is the value of the name=" " attribute of an image <map> element.
- ismap a Boolean attribute indicating that there is an ancestor <a href=" "/a> element which provides access to a server side image map.
- width=" " where there is a srcset=" " attribute but no sizes=" " attribute.
- height=" " where there is a srcset=" " attribute but no sizes=" " attribute.
- referrerpolicy=" " the referrer policy for fetches initiated by the element.
- decoding=" " a decoding hint to use when processing this image for presentation; it may take the values sync, async or, the default, auto.
- loading=" " indicates the policy for loading images that are outside the viewport; it may take to values eager or lazy.
- fetchpriority=" " sets priority for fetches.

An <img> element should be inside a <figure> element along with a <figcaption> element where it is the content rather than just an illustration to the content, such as,

```
<p>My home town is <img src="Halifax.png"
 alt="Halifax">.</p>
```

If the image is not specific to any of the content of the page but purely decorative, it should be placed in the style sheet.

Placing SVG images in an <img> element is discouraged because it offers no fallback image (Bah, 2011, p. 412); use the <object> element instead.

10.4 iframe

The <iframe> element represents a nested browsing context and must be an application, document or image but is primarily intended for text; it should have a srcdoc=" " attribute giving the HTML content of the page which the nested browsing context is to contain[14] and a src=" " attribute giving the address of the page which acts as a fallback for browsers which do not support the srcdoc=" " attribute. It must have the src=" " attribute if the itemprop=" " attribute is used. It may also have the attributes:

- name=" " a name for the nested browsing context which can be accessed in Javascript with window.<name>
- sandbox=" " takes any of the following space separated attributes

  - allow-downloads
  - allow-forms
  - allow-modals
  - allow-orientation-lock
  - allow-pointer-lock
  - allow-popups
  - allow-popups-to-escape-sandbox
  - allow-presentation
  - allow-same-origin
  - allow-scripts

[14] See https://html.spec.whatwg.org/#the-iframe-element for some cautions on using quotations and ampersands in the page.

  - `allow-top-navigation`

    OR
  - `allow-top-navigation-by-user-activation`

    OR
  - `allow-top-navigation-to-custom-protocols` if `allow-popups` is not specified

  each of which change aspects of the sandboxing; allowing both the attributes `allow-scripts` and `allow-same-origin` to be set negates the sandbox.

  The MIME type should be `text/html-sandboxed` but, since potentially hostile files with this MIME type can still be served from the server hosting the `<iframe>` element, they should always be served from a different domain to ensure that they benefit from all aspects of sandboxing.

- `allow` specifies the container policy that will be used
- `allowfullscreen` a Boolean attribute which allows the browsing context to `requestFullscreen()`; it is ignored if an ancestor `<iframe>` element does not have it set
- `allowpaymentrequest` a Boolean attribute which allows the browsing context to use the `PaymentRequest` interface to make payment requests
- `width=" "`
- `height=" "`
- `referrerpolicy=" "` the referrer policy for fetches initiated by the element
- `loading=" "` indicates the policy for loading images that are outside the viewport; it may take to values `eager` or `lazy`.

The ARIA attribute `role=" "` may take the values:

- `application` where it contains one or more focusable elements requiring user input, such as keyboard or gesture events; use `aria-labelledby=" "` if a visible label is present or `aria-label=" "` if not;
- `document` where it contains a child element that assistive technology users may want to browse in a reading mode and the relevant widget or application does not make it clear what its function is;
- `img` where it contains a collection of elements that form an image.

Placing SVG images in an `<iframe>` element is discouraged because it offers no fallback (Bah, 2011, p. 411-12); use the `<object>` element instead.

10.5 embed

The `<embed>` element offers an integration point for an external, non-HTML element; it must be an application, document or image.

It should normally have a `src=" "` attribute (and must have one if the `itemprop=" "` attribute is specified) and a `type=" "` attribute relevant to the `src=" "` attribute and may have `width=" "` and `height=" "` attributes; the `src=" "` attribute may have attributes which are passed as parameters to a plugin, in which case it is wise to have the element sandboxed.

In the absence of a `src=" "` attribute, the `type=" "` attribute should contain the MimeType by which the plugin is selected. It should not be a descendant of a media element or an `<object>` element that is not showing fallback content. It has no end tag. The ARIA attribute `role=" "` may take the values:

- `application` where it contains one or more focusable elements requiring user input, such as keyboard or gesture events; use `aria-labelledby=" "` if a visible label is present or `aria-label=" "` if not;
- `document` where it contains a child element that assistive technology users may want to browse in a reading mode and the relevant widget or application does not make it clear what its function is;

- img where it contains a collection of elements that form an image.

Placing SVG images in an <embed> element is discouraged because it offers no fallback (Bah, 2011, p. 411); use the <object> element instead.

10.6 object

The <object> element represents an external resource, which may be treated as an image, as a nested browsing context, or as an external resource to be processed by a plugin; it must be an application, document or image. It should have one or both of the attributes:

- data=" " a URL specifying the source; it must be present if the itemprop=" " attribute is used
- type=" " specifying the MIME type of the resource

and may have the attributes:

- name=" " a name for the nested browsing context which can be accessed in Javascript with window.<name>
- form=" " which associates the object with the id=" " attribute of its form owner, if any
- width=" "
- height=" "

The ARIA attribute role=" " may take the values:

- application where it contains one or more focusable elements requiring user input, such as keyboard or gesture events; use aria-labelledby=" " if a visible label is present or aria-label=" " if not;
- document where it contains a child element that assistive technology users may want to browse in a reading mode and the relevant widget or application does not make it clear what its function is;
- img where it contains a collection of elements that form an image.

The <object> element should normally be used to hold SVG images because, if the browser cannot render the SVG image, it will normally render any other content between <object> and </object> (Bah, 2011, pp. 410f), for example,

```
<object type="image/svg+xml" data="Blue_square.svg">

    <img src="Blue_square.png" alt="Blue square"/>

</object>
```

10.7 video

The <video> element is a media element used for playing videos or audio with captions; it must be an application and may take the following attributes:

- src=" " the address of the media resource
- crossorigin allows cross origin access; it may have one of the values:
  - anonymous
  - use-credentials

  if used, it is recommended that strict access precautions are taken to prevent port scans
- poster=" " provides the URL of an alternative image, normally a single frame from the video, to be displayed prior to playing or if the video cannot be played

- `preload=" "` is overridden if the `autoplay` attribute is set; it may take the attributes:
  - `none`
  - `metadata` loads the metadata but not the content
  - `auto` (the default) loads the source;

  this attribute gives a hint as to how much buffering may be needed
- `autoplay` a Boolean attribute which instructs the browser to start playing as soon as it can do so without needing to stop; n.b. it should be preferred to a script so that the user can block autoplay with a script such as:

```
var arVideos = document.getElementsByTagName('video');
for (var i = arVideos.length - 1; i >= 0; i--) {

    var elmVideo = arVideos[i]; elmVideo.autoplay = false;

}
```

(Pilgrim, 2010, p. 111)

- `playsinline` a Boolean attribute which indicates that video content should be displayed within the element's playback area
- `loop` a Boolean attribute which indicates that the media element should seek the start of the media resource when it reaches the end
- `muted` a Boolean attribute that, if set, controls the default state of the audio output
- `controls` a Boolean attribute which, when set, indicates that the browser should provide the necessary controls
- `width=" "` default 300 CSS pixels; supposed to be for use by legacy browsers but may be needed with modern browsers
- `height=" "` default 150 CSS pixels; supposed to be for use by legacy browsers but may be needed with modern browsers

It takes the ARIA attribute `role="application"`.

For further information on detecting and encoding video formats, see appendix D on page 79.

10.8 audio

The `<audio>` element is a media element used for playing sound; it must be an application and may take the following attributes:

- `src=" "` the address of the media resource
- `crossorigin` allows cross origin access; it may have one of the values:
  - `anonymous`
  - `use-credentials`

  if used, it is recommended that strict access precautions are taken to prevent port scans
- `preload=" "` is overridden if the `autoplay` attribute is set; it may take the attributes:
  - `none`
  - `metadata` loads the metadata but not the content
  - `auto` (the default) loads the source;

this attribute gives a hint as to how much buffering may be needed

- autoplay a Boolean attribute which instructs the browser to start playing as soon as it can do so without needed to stop; n.b. it should be preferred to a script so that the user can block autoplay if desired
- loop a Boolean attribute which indicates that the media element should seek the start of the media resource when it reaches the end
- muted a Boolean attribute that, if set, controls the default state of the audio output
- controls a Boolean attribute that, when set, indicates that the browser should provide the necessary controls.

It takes the ARIA attribute role="application".

Note: Use the <video> element where accessibility is an issue as this can provide alternatives to the audio channel.

10.9 track

The <track> element is a child of a media element intended to specify external timed text tracks; it takes the attributes:

- kind=" " which may have the attributes
  - subtitles the dialogue (the default), in which case the srclang=" " attribute must be specified; for where the language is not understood
  - captions the dialogue, sound effects and other audio information; for where hearing is difficult, for example, no or poor sound or a person is hard of hearing
  - descriptions of the video information; for use in the absence of vision, for example, no screen or a person is blind
  - chapters for use in navigating; not displayed
  - metadata for use from a script; not displayed
- src=" " the address of the media resource
- srclang=" " the language of the track
- label=" " a user readable title for the track; this must be unique, that is, there must be no other <track> element pointing to the same content; the lang=" " attribute may be used to specify the language of the title (as opposed to the language of the track)
- default a Boolean attribute indicating that this track is to be used in the absence of any other user preference; there must be no more than one track with this attribute set where the value of kind=" " is not empty or metadata.

It has no end tag.

10.10 map

A <map> element and its child <area> elements define an image map; it must have the attribute name=" " which must be a unique value which can be referenced by a usemap=" " attribute. The value of any id=" " attribute must be the same as the value of the name=" " attribute.

10.11 area

The <area> element is a child of a <map> or <template> element; it creates a hyperlink either

1. directly using the attributes:
   - href=" "
   - alt=" " a textual alternative to the image which may be omitted only if there is another <area> element which is a child element of the same <map> element and which contains a non-empty alt=" " attribute

   in which case the following attributes may be used:

- `target=" "` which activates a browsing context (see Other notes on attributes)
- `download` downloads a resource from a hyperlink; it may have as a value the filename to be given to the downloaded file; this must be chosen carefully to reflect the different conventions used by different filesystems
- `ping=" "` contains the URLs of resources that are interested in knowing if the user follows the hyperlink; authors are encouraged to use this rather than HTTP redirects or Javascript because it can be disabled by the user
- `rel=" "` make take the values:
  - `alternate` links to an alternate representation of the current document; if `hreflang=" "` is different from the root language of the document, it means that the linked content is a translation; if the `type=" "` attribute is not the same as the `type=" "` attribute of the parent element, the linked content will be an alternative format of the the current document.[15]
  - `author` links to a document describing the current section's author or her/his email address
  - `bookmark` creates a hyperlink to the nearest ancestor section
  - `external` indicates that the referenced document is on a different site
  - `help` creates a hyperlink to context sensitive help for the parent element
  - `license` creates a hyperlink to a licence document; the author must make clear the element to which it applies
  - `next` creates a hyperlink to the next document in the series to which the current document belongs
  - `nofollow` the link is not endorsed by the author/publisher of the current document
  - `noopener` the browsing context created by following this link is disowned
  - `noreferrer` requires the user agent not to send an HTTP Referer [*sic*] header with the link, preventing display of the URL from which this URL was accessed (that is, it prevents use of the Javascript `d.referrer` document property)
  - `opener` creates an auxiliary browsing context, i.e. one that has `"_blank"` as a `target` attribute value
  - `prev` creates a hyperlink to the previous document in the series to which the current document belongs; the legacy value `previous` is treated in the same way
  - `search` creates a hyperlink to a resource which can be used to search the current document
  - `tag` creates a hyperlink to an address which gives a tag describing the current document
- `referrerpolicy=" "` for any fetches initiated by the element
- `itemprop=" "` see Microdata on page 52
- `role="link"` if the content of the element does not make it clear what its function is

or

2. by pointing to an area on an image map; it may then have the following attributes:
   - `coords=" "` mandatory except where `shape="default"`
   - `shape=" "` may have one of the attributes:
     - `circle` where `coords=(x,y,r)` represents the location of the centre in Cartesian coordinates and the radius of the circle

[15] See also the footnote on page 8.

- default = rect whose coordinates fill the available content
- poly where coords=" " contains a minimum of three paired Cartesian coordinates
- rect where coords=(x,y,w,h) contains two Cartesian coordinates giving the top left corner along with the width and height in pixels.

It has no end tag.

11 Foreign elements

11.1 math

The <math> element contains MathML elements such as <mi>, <mo>, <mn>, <ms> and <mtext>.

It takes the ARIA attribute role="math". If the element is navigable, use aria-labelledby=" " if a visible label is present or aria-label=" " if not; if the element is not navigable, use aria-label=" " to hold either the content or the name of the expression.

11.2 svg

The <svg> element contains SVG elements.

Note that an SVG image in an <img>, <iframe>, <embed>, <object> or <video> element or an <input> element with type="button" must have the width=" " and height=" " attributes specified.

But note also the advice from Bah (2011, pp. 410f) that the <object> element should normally be used to hold SVG images because, if the browser cannot render the SVG image, it will normally render any other content between <object> and </object>.

It normally takes the ARIA attribute role="img".

12 Tabular data

12.1 table

The <table> element represents data in more than one dimension; it should not be used for layout purposes for which CSS should be used. It may take the Boolean attribute sortable and the ARIA attributes

- role="grid" where some or all cells in the grid are focusable by using methods of two-dimensional navigation, such as directional arrow keys; the aria-readonly attribute will be set to true if it has the Boolean attribute sortable
- role="table" where none of the cells are interactive.

In the absence of a <caption> element give the table a title with:

- aria-labelledby=" " if a visible label is present or
- aria-label=" " if not.

12.2 caption

The <caption> element is the first child of a <table> element which is not the main element in a <figure> element, when the <figcaption> element should be used to give the table a caption.

The <caption> element should take the ARIA attribute aria-labelledby on the parent table. However, if a caption contains content that serves as both a name and description for its parent, authors MAY instead set aria-labelledby to reference an element within the caption that contains a concise name, and set aria-describedby to reference an element within the caption that contains the descriptive content.

12.3 colgroup

The <colgroup> element either contains one or more <col> or <template> elements or has a span= attribute whose value is not 0; it is a child of a <table> element, coming after any <caption> element and before any <thead>, <tbody>, <tfoot> or <tr> elements.

12.4 col

The <col> element represents one or more columns within a <colgroup> element; if it has the span= attribute specifying more than one column, its parent <colgroup> element must not have the span= attribute. It has no end tag.

12.5 thead

The <thead> element represents a block of rows, not overlapping any other block of rows, consisting of the column headers, following any <caption> or <colgroup> element and before any <tbody>, <tfoot> or <tr> elements. There can only be one <thead> element in a <table> element.

12.6 tbody

The <tbody> element represents a block of rows, not overlapping any other block of rows, within a <table> element which has no <tr> child elements; it comes after any <caption>, <colgroup> or <thead> elements.

It normally takes the ARIA attribute role="rowgroup".

12.7 tr

The <tr> element represents a row of cells in a <thead>, <tbody> or <tfoot> element or in a <table> element following any <caption>, <colgroup> or <thead> element if there is no <tbody> element, which in this case is implied.

It takes the ARIA attribute role="row" if the content of the element does not make it clear what its function is and it is a descendant of a focusable element with the attribute treegrid.

12.8 th

The <th> element is a child of a <tr> element and represents header cells in a <tr> element; it must not contain any sectioning or heading elements or <header> or <footer> elements; it may take the attributes:

- colspan= whose value should not cause any overlap
- rowspan= whose value should not cause any overlap; 0 = all the remaining rows in the group
- headers=" " contains the space separated values of the id=" " attributes of the <th> elements, other than itself, relevant to itself
- scope=" " which may take one of the attributes:
  - row may apply to the remaining cells in this row in which case it takes the ARIA attribute role="rowheader" if the

parent element has the attribute `role="grid"` and the content of the element does not make it clear what its function is;

  - `col` may apply to the remaining cells in this column in which case it takes the ARIA attribute `role="columnheader"` if the parent element has the attribute `role="row"` and the element does not make it clear what its function is;
  - `rowgroup` applies to the remaining cells in this row group but only if the element is a descendent of a `<tbody>` element
  - `colgroup` applies to the remaining cells in this column group but only if the element is a descendent of a `<colgroup>` element
  - `auto` (the default) based on context

- `abbr=" "` an abbreviation of the heading text which can be referenced from elsewhere.

Where `scope` is not set to `row` or `col` and the element's parent's role is `grid`, it takes the ARIA attribute `role="gridcell"` if the element does not make it clear what its function is.

A header cell may overlap column groups but not row groups.

Text in a `<th>` element is normally centred and in bold.

12.9 td

The `<td>` element represents a data cell in a `<tr>` element; it may take the attributes:

- `colspan=` whose value should not cause any overlap
- `rowspan=` whose value should not cause any overlap
- `headers=" "` contains the space separated values of the `id=" "` attributes of the `<th>` elements relevant to itself.

It may take the following ARIA attributes where its parent has the ARIA attribute `role="row"` and the element does not make it clear what its function is:

- `role="gridcell"` where the element is focusable, editable, and selectable or
- `role="cell"` in any other case.

A data cell may overlap column groups but not row groups.

12.10 tfoot

The `<tfoot>` element represents a block of `<tr>` elements, not overlapping any other block of rows, containing the column summaries following any `<caption>`, `<colgroup>`, `<thead>`, `<tbody>` or `<tr>` elements. There can only be one `<tfoot>` element in a `<table>` element.

13 Forms

The `<form>` element offers a way of obtaining user data; each data item is normally collected by an `<input>` element described by and enclosed in a `<label>` element, both contained in a `<p>` element.

Form elements may take the following attributes, where appropriate:

- `autocomplete=" "` may take the values `on` (the default), `off` or a variety of other values; see page on page 38
- `autofocus` a Boolean attribute which, when set, enables the user to begin typing immediately in the selected element, for

example,

```
<input name="q" autofocus>
<input type="submit" value="Search">
```

see autofocus on page 39

- `dirname=" "` records the direction in which the user entered the text and appends that to the input in the form `value=direction`, for example, if the value of `dirname=" "` was `"entry"`, `entry=ltr`
- `formaction=" "` for use only where `type="submit"`; specifies a URL; in its absence, the URL specified in the `action` attribute of the `<form>` element is used; for use only where `type="submit"`
- `formenctype=" "` for use only where `type="submit"`; make take the attributes
  - `application/x-www-form-urlencoded`
  - `multipart/form-data`
  - `text/plain`

  and the corresponding state; in its absence, the encoding specified in the `enctype` attribute of the `<form>` element is used; for use only where `type="submit"`
- `formmethod=" "` for use where `type="image"` or `type="submit"`; may take the attributes:
  - `get` corresponding to HTTP GET (the default)
  - `post` corresponding to HTTP POST
  - `dialog` closing the dialog box but submitting nothing

  In its absence, the encoding specified in the `method` attribute of the `<form>` element is used
- `formnovalidate` for use only where `type="submit"`; a Boolean attribute which, if set, indicates that the element is not to be validated; disabled if `novalidate` is set in the `<form>` element; for use only where `type="submit"`; used to enable a user to save a partially completed form
- `formtarget=" "` for use only where `type="submit"`; specifies a browsing context; in its absence, the `target` attribute of the `<form>` element is used; for use only where `type="submit"`
- `inputmode=" "` may take the values
  - `verbatim` non-pose alphanumeric Latin script, e.g. user names, passwords
  - `latin` free form Latin script for human to computer interaction, e.g. for searches
  - `latin-name` free form Latin script for human to computer interaction, including writing aids
  - `latin-prose` free form Latin script prose for human to human interaction, including writing aids
  - `full-width-latin` free form Latin script prose for human to human interaction in the user's secondary language, including writing aids
  - `kana` Roman input of Japanese text
  - `katakana` for Japanese text
  - `numeric` for numeric date other than numbers for which `<input type="number">` should be used
  - `tel` deprecated: use `<input type="tel">`
  - `email` deprecated: use `<input type="email">`
  - `url` deprecated: use `<input type="url">`
- `maxlength=` specifies the maximum code-point length of the text in a text input element when it is submitted
- `name=" "` a unique name which can be accessed in Javascript
- `required` a Boolean attribute which, when set, requires the user to enter a response.

`autocomplete`

If missing, it inherits the default state of the `<form>` element which, if not otherwise specified, is "`on`".

If present, the `autocomplete` attribute gives hints to the browser's autofill mechanisms; `autocomplete="off"` turns off autocompletion to ensure that sensitive data is entered by the user.

If present and not `off`, it may take the values `on` (the default) or:

1. space separated values of the form `section-*` with, optionally, the values `shipping` or `billing`

   or

2. one of the tokens where `<input type="text">` unless otherwise stated[16]

   - `name` full name or, as separate values,
     - `honorific-prefix` Mr Ms. etc.
     - `given-name`
     - `additional-name` any names other than `given-name` and `family-name`
     - `family-name`
     - `honorific-suffix` PhD
   - `nickname`
   - `username`
   - `new-password` where `<input type="password">`
   - `current-password` where `<input type="password">`
   - `one-time-code`
   - `organization-title` job title
   - `organization`
   - `street-address` with `<textarea>` or `<select multiple>` elements or, as separate values,
     - `address-line1`
     - `address-line2`
     - `address-line3`
   - `address-level4`
   - `address-level3`
   - `address-level2` locality
   - `address-level1` state, province, region, county
   - `country` ISO3166 country code
   - `country-name` country
   - `postal-code`, if not CEDEX which should be included in `address-level2` field
   - `cc-name` full name as on a payment instrument, consisting of
     - `cc-given-name`
     - `cc-additional-name`
     - `cc-family-name`
   - `cc-number` card number
   - `cc-exp` where `<input type="month">` expiry date of card, consisting of
     - `cc-exp-month` where `<input type="number">`
     - `cc-exp-year` where `<input type="number">`
   - `cc-csc` security code
   - `cc-type` card type
   - `transaction-currency` ISO 4217 currency code
   - `transaction-amount` floating point number
   - `language` BCP 47 language tag, for example, en, fr
   - `bday` where `<input type="date">` birthday in full, consisting of:
     - `bday-day` where `<input type="number">`
     - `bday-month` where `<input type="number">`

[16] In practice, in many cases `<input type="search">` will also work and in most cases `<input type="text">` will work as well as the alternative given.

        - bday-year where <input type="number">
    - sex in free form text
    - url where <input type="url">
    - photo where <input type="url">

   plus, optionally,

3. one of the following strings:
    - home, meaning the field is for contacting someone at their residence
    - work, meaning the field is for contacting someone at their workplace
    - mobile, meaning the field is for contacting someone regardless of location
    - fax, meaning the field describes a fax machine's contact details
    - pager, meaning the field describes a pager's or beeper's contact details
4. followed by one of the following autofill field strings:

- tel where <input type="tel"> full ISDN telephone number, consisting of:
    - tel-country-code
    - tel-national consisting of:
        - ∗ tel-area-code
        - ∗ tel-local consisting of:
            - · tel-local-prefix
            - · tel-local-suffix
- tel-extension
- email where <input type="email">
- impp where <input type="url"> messaging protocol.

autofocus

When set, the autofocus attribute enables the user to begin interacting immediately with the selected element where it is

- an <a> element with the href=" " attribute
- a <link> element with the href=" " attribute
- a <button> element that is not disabled
- an <input> element whose type=" " attribute is not hidden or disabled
- a <select> element that is not disabled
- a < textarea> element that is not disabled
- an element with a draggable attribute set, if that would enable the user agent to allow the user to begin a drag operations for those elements without the use of a pointing device
- an editing host
- a shape that is generated for an < area> element
- part of a <details> element's rendering.

A script for autofocus with fallback

This script requires the scripts in the <head> element of the simple HTML framework in table 1 on page 9 to have run.

```
<form name="f">

    <input id="q" autofocus>
       <script>$(document).trigger('autofocus_ready'); </script>
    <input type="submit" value="Go">
</form>
```

13.1 form

The `<form>` element contains a collection of form associated elements; it must take the attribute

- `name=" "` a unique name for the form which can be accessed in Javascript with `document.<name>`

and may take the following attributes:

- `accept-charset=" "` specifies the character encoding of submitted material
- `action=" "` specifies a URL to use for form submission
- `autocomplete=" "` see `autocomplete` on page 38
- `enctype=" "` specifies the data set encoding of the submitted material; it may take take the attributes
  - `application/x-www-form-urlencoded` (the default)
  - `multipart/form-data`
  - `text/plain`

  and the corresponding state.
- `method= " "` may take the attributes:
  - `get` corresponding to HTTP GET (the default)
  - `post` corresponding to HTTP POST
  - `dialog` closing the dialog box but submitting nothing
- `novalidate` a Boolean attribute which, if set, indicates that the form is not to be validated; used to enable a user to save a partially completed form
- `target=" "` specifies a browsing context (see Other notes on attributes)
- `rel=" "` make take the values:
  - `noopener` the browsing context created by following this link is disowned
  - `noreferrer` requires the user agent not to send an HTTP Referer [*sic*] header with the link, preventing display of the URL from which this URL was accessed (that is, it prevents use of the Javascript `d.referrer` document property)
  - `opener` creates an auxiliary browsing context, i.e. one that has `"_blank"` as a `target` attribute value

A `<form>` element cannot appear inside another `<form>` element or in any phrasing element (see appendix B.5 on page 73) because it implies the end tag of any phrasing element.

It takes the ARIA attribute `role="form"`. If the purpose of the element is unclear, use `aria-labelledby=" "` if a visible label is present or `aria-label=" "` if not.

13.2 label

The `<label>` element is normally a child of a `<p>` element; it provides a caption for an input method which is either:

- its first descendant input method element or:
- the element whose `id=" "` attribute value matches the value of the `for=" "` attribute of the `<label>` element

13.3 input

The `<input>` element is an input method which may have a button, combobox, slider, spinbox or textbox role depending on its attributes which may include:

- `accept=" "` where `type="file"`; contains a hint, in the form of a case-insensitive comma separated list, as to acceptable file-types whether as a MIME type, with one of the forms `audio/*`, `video/*` or `image/*` or with a string beginning with `.` to indicate acceptable file endings
- `alt=" "` where `type="image"`; provides a textual input for those who cannot access the image button
- `autocomplete=" "` where `type=" "` is anything except `checkbox` or `radio`; see `autocomplete` on page 38
- `checked` where `type="checkbox"` or `type="radio"`
- `dirname=" "` where `type="text"` or `type="search"`; see Forms on page 36
- `disabled`; the property `aria-disabled` is `true` if `disabled` is `true`
- `form=" "` explicitly associates the `<input>` element with the `id=" "` attribute of its form owner
- `formaction=" "` where `type="submit"` or `type="image"`; see Forms on page 36
- `formenctype=" "` where `type="submit"` or `type="image"`; see Forms on page 36
- `formmethod=" "` where `type="submit"` or `type="image"`; see Forms on page 36
- `formnovalidate` where `type="submit"` or `type="image"`; see Forms on page 36
- `formtarget=" "` where `type="submit"` or `type="image"`; see Forms on page 36
- `height=` where `type="image"`
- `list=" "` where `type=" "` is anything except `hidden`, `password`, `checkbox` or `radio`; if the `multiple` attribute is set, takes the value of the `id=" "` of a `<datalist>` element in the same document
- `max=` where `type=" "` is a date, time, number or range; with dates, times must be a string valid for the `type=" "` attribute; with ranges the default is 100
- `maxlength=` for use with any text field including passwords; specifies the maximum code point length of the entry
- `min=` where `type=" "` is a date, time, number or range; with dates, times must be a string valid for the `type=" "` attribute; with ranges, the default is 0; it also defines the step base
- `minlength=` for use with any text field including passwords; specifies the minimum code point length of the entry
- `multiple` a Boolean attribute for use where the `list=" "` attribute is set or `type="email"` or `type="file"`; enables comma separated email addresses filenames or selections from a range or from a `<datalist>` element pointed to by the `list=" "` attribute to be submitted
- `name=" "` provides a key for the value of an `<input>` element, each key within a particular form should be unique except where `type="radio"` when all elements in a radio group must have the same key
- `pattern=" "` for use with any text field including passwords; permits the use of Javascript regular expressions, for example, to enforce patterns of data entry in URL, email or telephone fields; there must be a `title=" "` attribute specifying for the user the available patterns
- `placeholder=" "` contains the text that will appear in a textbox where `value=" "` is empty or absent; for use with any text field including passwords and where `type="number"`, for example,

  ```
  <input name="q" placeholder="Search
  Bookmarks and History">
  <input type="submit" value="Search">
  ```

  It is not an alternative to the `<label>` element; the value of the `title=" "` attribute should be displayed after the user has commenced an entry.

- `readonly` a Boolean attribute for use where `type=" "` is anything except `hidden`, `checkbox`, `radio`, `range` or `color`; the property `aria-readonly` is `true` if `readonly` is `true`; unlike an element whose the `disabled` attribute is set, users can interact with an element whose `readonly` attribute is set
- `required` a Boolean attribute for use where `type=" "` is anything except `hidden`, `range` or `color`; if applied to one `<button>` element, it applies to all those in the button group; the property `aria-required` is `true` if `required` is `true`
- `size=" "` for use with text including passwords; the default is 20 characters
- `src=" "` for use with image buttons
- `step=" "` where `type=" "` is a date, time, number or range; the default datetime and time steps, always expressed in seconds, are 60 seconds and the default date, week, month and number steps are 1; a number step must be an integer unless the value of `min=` is not an integer; where `step="any"` there is no step value
- `title=" "` specifies for the user the available patterns when the `pattern=" "` attribute is set
- `type=" "` which may take the attributes:
  - `hidden` for input not to be provided by the user; the property `aria-hidden` is `true` if `hidden` is `true`
  - `text` one line of plain text (the default and the fallback for browsers that do not recognise one or more of the other types); it can normally be omitted; the ARIA attribute `role="textbox"` may be set if there are no suggestions as to sources; the AREA attribute `role="combobox"` may be set if there are suggestions as to sources and the `list` attribute is specified; in this case the property `aria-owns` holds this value
  - `search` generates a textbox for one line of plain text with an integral clear contents button; the following ARIA attributes may be set:
    - `role="textbox"` if there are no suggestions as to sources;
    - `role="searchbox"` if there are suggestions as to sources;
    - `role="combobox"` if there are suggestions as to sources and the `list` attribute is specified; in this case the property `aria-owns` holds this value
  - `tel` intended for telephone numbers; the ARIA attribute `role="textbox"` may be set if there are no suggestions as to sources; the AREA attribute `role="combobox"` may be set if there are suggestions as to sources and the `list` attribute is specified; in this case the property `aria-owns` holds this value
  - `tel-national` where only telephone numbers from a particular country are acceptable
  - `url` the ARIA attribute `role="textbox"` may be set if there are no suggestions as to sources; the AREA attribute `role="combobox"` may be set if there are suggestions as to sources and the `list` attribute is specified; in this case the property `aria-owns` holds this value; it is worth converting all url input elements to this type because it prompts some mobile phone browsers to offer a restricted keyboard containing alphnumerics, `/` and `.` only; those that do not recognise it will default to `text` anyway
  - `email` accepts more than one email address if the `multiple` attribute is set; the ARIA attribute `role="textbox"` may be set if there are no suggestions as to sources; the AREA attribute `role="combobox"` may be set if there are suggestions as to sources and the `list` attribute is specified; in this case the property `aria-owns` holds this value; it is worth converting all email input elements to this type because it prompts some mobile

phone browsers to offer a restricted keyboard containing alphnumerics, `@` and `.` only; those that do not recognise it will default to `text` anyway

  - `password` echoes dots instead of characters as the user types; the ARIA attribute `role="textbox"` may be set
  - `date` generates a date picker using `yyyy-mm-dd`
  - `month` generates a month picker using `yyyy-mm`
  - `week` generates a week picker using `yyyy-W`*n*
  - `time` generates a time picker using `hh:mm:ss`
  - `datetime-local` generates a date/time picker based on local time using `yyyy-mm-dd [T]hh:mm:ss`
  - `number` intended for the use of floating point numbers in spinboxes; it is not intended for number strings such as credit card numbers but otherwise it is worth converting all number input elements to this type because it prompts smartphone browsers to offer a restricted keyboard containing numerals and symbols only or a spinbox; those that do not recognise it will default to text anyway and display any `value=" "` attribute that has been set. It may take the ARIA attribute `role="spinbox"` in which case `aria-valuemax` is set to the maximum value, `aria-valuemin` to the minimum value and `aria-valuenow` to the number entered
  - `range` intended for sliders; if a browser does not render it as a slider, it will default to text anyway; if any of the values are unspecified, it takes the default values for the `min=`, `max=` and `value=" "` attributes; it may take the ARIA attribute `role="slider"` if the attribute `multiple` is not specified in which case `aria-valuemax` is set to the maximum value, `aria-valuemin` to the minimum value and `aria-valuenow` to the value selected
  - `color` an sRGB colour in 8 bit red, green and blue; without a value, it is intended to generate a colour picker but this is as yet unsupported
  - `checkbox` a set of zero or more values from a predefined list; the property `aria-checked` may be `mixed` if its state is indeterminate, `true` or `false`; it takes the ARIA attribute `role="checkbox"`
  - `radio` an enumerated value; there must always be more than one such `<input>` element and each element in the group must have the same value for the `name=" "` attribute; the property `aria-checked` may be `true` or `false`; it takes the ARIA attribute `role="radio"`.
  - `file` opens an "open file" dialogue in order to upload a file or more than one file if the `multiple` attribute is set; the `accept=" "` attribute may be used to limit the files which are acceptable
  - `submit` makes the `<input>` element a command initiating submission; it may take the ARIA attribute `role="button"`
  - `image` makes selecting a coordinate within the image a command initiating submission; it needs a `src=" "` and an `alt=" "` attribute providing a textual input for those who cannot access the image; it may also have dimension attributes; it normally takes the ARIA attribute `role="button"`.
  - `reset` makes the `<input>` element a command resetting the entire form; it may take the ARIA attribute `role="button"`
  - `button` requires a `value=" "` attribute as a label; it makes the `<input>` element a command; the ARIA attribute `role=" "` may take one of the values:
    * `button`; the default

- `value=" "` may contain:
  - the text that accompanies the input method;; this must be

consistent with any label where `type="submit"` and may be a label where `type="button"`

  - the URL where `type="url"`
  - a comma separated list of email addresses where `type="email"`
  - a valid global date and time if the `type="datetime"`; the attributes `min=` and `max=` may also hold a global date-time values in which case any value in `step=" "` represent seconds.

    Similar considerations apply to the other date and time related values.
  - a floating point number where `type="number"` or `type="range"`; in the latter case, the attributes `min=" "` and `max=` may also hold floating point numbers in which case the value of `step=" "` defaults to `1` unless `min=` is not an integer.
  - an sRGB colour where `type="color"`
  - the value associated with an element where `type="checkbox"` or `type="radio"`
- `width=` where `type="image"`.

See also section 14.4 on page 48.

13.4 button

The `<button>` element always defines a command; it may take the following attributes:

- `autofocus` see autofocus on page 39
- `disabled`; the property `aria-disabled` is `true` if `disabled` is `true`
- `form=" "` explicitly associates the `<button>` element with the `id=" "` attribute of its form owner
- `formaction=" "` where `type="submit"`, see Forms on page 36
- `formenctype=" "` where `type="submit"`, see Forms on page 36
- `formmethod=" "` where `type="submit"`, see Forms on page 36
- `formnovalidate` where `type="submit"`, see Forms on page 36
- `formtarget=" "` where `type="submit"`, see Forms on page 36
- `name=" "` a name for the button
- `popovertarget=" "` takes the ID of an element with a `popover` attribute in the same tree as the `<button>` with the `popovertarget` attribute.
- `popovertargetaction=" "` takes the attributes:
  - `toggle` shows or hides the targeted popover element
  - `show` shows the targeted popover element
  - `hide` hides the targeted popover element.
- `type=" "` make take the attributes:
  - `submit` submits the form (the default attribute)
  - `reset` resets the form
  - `button` does nothing but may be associated with an action, for example,

```
<button type=button onclick="alert('This 15-20 minute piece was composed by George Gershwin.')">

    Show hint

</button>
```

- `value=" "` provides the `<button>` element's label

A `<button>` element cannot contain a `<textarea>` element.

It only takes the ARIA attribute `role="button"` in the unlikely case that the content of the element does not make it clear that it is a `<button>` element.

See also section 14.4 on page 48.

13.5 select

The `<select>` element is an input method which controls the selection of options from a number of child `<option>` elements or `<option>` elements within a child `<optgroup>` element; it may have the attributes:

- `autocomplete=" "` may take the values `on` (the default), `off` or a variety of other values; see page on page 38
- `disabled`; the property `aria-disabled` is `true` if `disabled` is `true`
- `form=" "` explicitly associates the `<select>` element with the `id=" "` attribute of its form owner
- `multiple` a Boolean attribute which permits the selection of multiple options; when it is omitted, only one child `<option>` element may be selected; selecting another `<option>` element deselects all the others; if `it` is set and the items within the list are static, the property `aria-multiselectable` will be set to `true`; otherwise it will be set to `false` even there is more than one choice.
- `name=" "` a name for the group of form associated elements
- `required` a Boolean attribute indicating that the user must select an option; if this is `true`, the property `aria-required` will be set to `true`
- `size` limits the number of options to show the user; the default is `1` unless `multiple` is set when it is `4`.

Where the `<select>` element only contains `<option>` elements and the `multiple` attribute is not set but the `required` option is, the first `<option>` element will be designated as the placeholder option.

It normally takes the ARIA attribute `role="listbox"`.

13.6 datalist

The `<datalist>` element contains a number of `<option>` elements;[17] it is the child of a `<label>` element whose `<input>` element has the `list` attribute.

It takes the ARIA attribute `role="listbox"` provided the property `aria-multiselectable` is set to `false`.

13.7 optgroup

The `<optgroup>` element is a child of a `<select>` element and contains a group of `<option>` elements; it must have the `label=" "` attribute giving the name of the group of `<option>` elements and may have the `disabled` attribute.

13.8 option

The `<option>` element may be a child of a `<select>`, an `<optgroup>` or a `<datalist>` element; it may have the attributes:

- `disabled`; the property `aria-disabled` is `true` if `disabled` is `true`
- `label=" "` a label for the option
- `selected` whether the option is selected by default
- `value=" "` a value for the option

Where it is an element in a `<select>` element, it is a command of type `radio` if the `<select>` element has no `multiple` attribute and type `checkbox` if it does.

[17] These may be wrapped in a `<select>` element for the benefit of browsers which do not support the `<datalist>` element.

Where it is an element in a <select> element with no multiple attribute or in an <optgroup>, a <datalist> or a <select> element with the multiple attribute, it may take the ARIA attribute role="option" where it is contained in, or owned by, an element with the role listbox. In this case, where it is in an <optgroup>, the property aria-selected will implicitly be false.

See also section 14.4 on page 48.

13.9 textarea

The <textarea> element provides a textbox for the entry of text; it may have the attributes:

- autocomplete see autocomplete on page 38
- cols= specifies the character width of the text area (the default is 20)
- dirname=" " see Forms on page 36
- disabled; the property aria-disabled is true if disabled is true
- form=" " explicitly associates the <textarea> element with the id=" " attribute of its <form> element
- maxlength= specifies the maximum code-point length of the text when it is submitted
- minlength= specifies the minimum code-point length of the text when it is submitted
- name=" " a unique name which can be accessed in Javascript with form.<name>
- placeholder=" " if present, contains a short hint that will appear in the textbox; for longer content use the title=" " attribute; it should not be used as an alternative to a <label> element
- readonly a Boolean attribute; the property aria-readonly is true if readonly is true
- required a Boolean attribute; the property aria-required is true if required is true
- rows= specifies the number of rows in the text area (the default is 2)
- wrap=" " takes the values soft (the default), in which case no newline characters are added when the text is submitted, or hard in which case the cols= attribute must be specified and the user agent should add newline characters when it is submitted.

It may not be inside a <button> element.

It normally takes the ARIA attribute role="textbox" with the property aria-multiline set to true.

13.10 output

The <output> element contains the result of a calculation or user action; it may take the attributes:

- for=" " specifies the value of the id=" " of the element with which the <output> element is associated
- form=" " explicitly associates the <output> element with the id=" " attribute of its form owner, if any
- name=" " a name for the <output> element.

It takes the ARIA attribute role="status".

13.11 progress

The <progress> element has a progress bar role; it may have the attribute:

- max= a floating point number value representing the completion of the task

  and, if it is a determinate progress bar, the attribute

- value= a floating point number value representing the progress made, normally generated by a script

The progress bar will be created using a script or other embedded code.

It normally takes the ARIA attribute `role="progressbar"` and, if the progress bar is determinate, `aria-valuemax` will be set to the maximum value, `aria-valuemin` to the minimum value and `aria-valuenow` to the current value.

N.B. The `<progress>` element should not be used as a gauge for which the `<meter>` element is more appropriate.

13.12 meter

The `<meter>` element contains a scalar measurement within a known range; it must have the attribute:

- `value=` a floating point number value representing the measurement made, normally generated by a script

and it may have the attributes:

- `min=` the lower bound of the measurement (the default is 0)
- `max=` the upper bound of the measurement (the default is 1)
- `low=` specifies the low boundary of the range (the default is `min`)
- `high=` specifies the high boundary of the range (the default is `max`)
- `optimum=` specifies the optimum point on the range (the default is the midpoint).

The `<meter>` element may be used in contexts where machine-readable data is used.

It may take the ARIA attributes:

- `role=alert` where it displays important, and usually time-sensitive, information or
- `role=status` where it displays advisory information not important enough to justify an `alert`.

Note: the `<meter>` element should not be used to record progress for which the `<progress>` element is more appropriate.

13.13 fieldset

The `<fieldset>` element contains, optionally, a `<legend>` element followed by a group of form control elements grouped under a name; it may take the attributes:

- `disabled` the property `aria-disabled` is `true` if `disabled` is `true`
- `form=" "` explicitly associates the `<fielset>` element with the `name=" "` attribute of its form owner
- `name=" "` a unique name which can be accessed in Javascript with `form.<name>`.

It takes the ARIA attribute `role="group"`.

13.14 legend

The `<legend>` element must be the first child of a `<fieldset>` element and contains its caption. It does not inherit the `disabled` attribute from its parent element and so always displays the caption.

14 Interactive elements

14.1 details

The `<details>` element enables the user to obtain further information; it normally contains a `<summary>` element acting as a legend for the details. It may have the attribute

- `open` a Boolean attribute which may be set if the details are to be shown to the user along with the summary; in this case the `aria-expanded` property is set to `true`.

It is not appropriate for material which should go in footnotes.

14.2 summary

The `<summary>` element contains a summary, caption or legend for a `<details>` element.

14.3 dialog

A `<dialog>` element contains something with which the user may interact. It should have the attribute `open`, a Boolean attribute specifying whether it is open to the user; it should always be `true` but, if it is not present, the property `aria-hidden` is set to `true`.

It normally takes the ARIA attribute `role="dialog"`; use `aria-labelledby=" "` if a visible label is present or `aria-label=" "` if not.

14.4 A note about commands

An element defines a command if it is:

- an `<a>` element with an `href=" "` attribute that is neither `disabled` nor `checked`
- a `<button>` element that is neither `disabled` nor `checked`
- an `<input>` element whose `type=" "` attribute is `submit`, `reset`, `image`, `button`, `radio` or `checkbox` and where the other attributes relevant to each type are consistent with that type
- an `< option>` element which is a child of a `<select>` element and whose `value=" "` attribute, if present, is not empty
- a `< legend>` element, where there is no other `<label>` or `<legend>` element within the `< fieldset>` element, whose `accesskey=" "` attribute is not empty and which is not `checked`
- any other element whose `accesskey=" "` attribute is not empty and which is neither `disabled` nor `checked`.

15 Scripting

Authors are encouraged to use scripts only when there is no other alternative.

15.1 script

The `<script>` element may take the following attributes:

- `src=" "` URL for a source file; if there is one, the following attributes may also be applied:
  - `async` a Boolean attribute indicating how a script should be evaluated; must not be used with classic scripts where the `src=" "` attribute is not present
  - `defer` a Boolean attribute indicating how a script should be evaluated; must not be used with classic scripts where the `src=" "` attribute is not present or with module scripts

  In the absence of both these attributes, the script is executed before the rest of the page is parsed.

  Where the `src=" "` attribute is defined, any other elements, including a script, in the `<script>` element are ignored; therefore, `<script>` elements containing scripts should not contain the `src=" "` attribute.
- `type=" "` the default value is `text/javascript`; it must be specified if it is a variant form of ECMAScript or Javascript but cannot be any of the plain text or data types.
- `nomodule` a Boolean attribute which prevents a script from being executed in user agents that support module scripts.
- `crossorigin` allows cross origin access; it may have one of the values:
  - `anonymous`
  - `use-credentials`

  if used, it is recommended that strict access precautions are taken to prevent port scans
- `integrity=" "` provides integrity metadata; must not be specified when embedding a module script or when the `src=" "` attribute is not present
- `referrerpolicy=" "` for any fetches initiated by the element
- `blocking` a Boolean attribute indicating whether the element is potentially render-blocking
- `fetchpriority=" "` sets the priority for fetches.

The property `aria-hidden` will be set to `true`.

15.2 noscript

The `<noscript>` element may be a child of the `<head>` element or any phrasing content element (section B.5 on page 73) provided there is no ancestor `<noscript>` element. It may contain a `<link>` , `<style>` or `<meta>` elements where it is a child of a `<head>` element. It may be used as a fallback where a script cannot be executed but, for legacy reasons, it is best avoided; if used, the property `aria-hidden` will be set to `true`.

15.3 template

The `<template>` element is a child of an element used to declare fragments of HTML that can be cloned and inserted into the element by a script. It may be the child of a `<colgroup>` element which does not have the `span` attribute.

The property `aria-hidden` will be set to `true`.

15.4 slot

The `<slot>` element defines a slot, typically used in a shadow tree that you can fill with your own markup, which lets you create separate DOM trees and present them together, to represent its assigned nodes. It may take the attribute `name=" "` being the name of the slot.

15.5 canvas

The `<canvas>` element provides a resolution dependent bitmap canvas to enable scripts to render visual images on the fly; it may have `<a>` elements, `<img>` elements with the `usemap` attribute set, `<button>` elements, `<input>` elements where `type="checkbox"` or `type="radio"`, `<select>` elements with the `multiple` attribute set, sortable `<th>` elements or elements with the `tabindex` attribute set as descendants; however, it should not be used to render images for which there are more suitable elements.

It may have one of the following `context modes`:

- `2d` in which case it always has two attributes:
  - `width=" "` which defaults to 300
  - `height=" "` which defaults to 150

  Whenever the width and height content attributes are set, removed, changed, or redundantly set to the value they already have, then the user agent must perform the action that corresponds with the canvas element's `context mode`.
- `webgl` or `webgl2` as defined in the WebGL specifications.
- `webgpu` as defined in WebGPU.
- `bitmaprenderer` if the context's bitmap mode is set to blank, run the steps to set an `ImageBitmapRenderingContext`'s output bitmap, passing the canvas element's rendering context.
- `placeholder` do nothing.
- `none` do nothing.

The `<canvas>` element should not be used for presentational purposes for which a style sheet should be used.

To enable a script to access the `<canvas>` element, add the `id=" "` attribute, for example,

```
<canvas id="a">
```

creates the variable `a_canvas` which can be located in the DOM with

`var a_canvas=document.getElementById("a");` and can be manipulated with `<id>_context.<instruction>`.

15.5.1 Canvas contexts

Every canvas can have one associated primary context called using, for example:

```
var a_context=a_canvas.getContext("<contextId>"[,<options>]);
```

where `contextId` may be

- `2d` (section 15.5.2 on the following page)
- `webgl/webgl2`, which supports a 3D context (section 15.5.3)
- `bitmaprenderer`, which runs the steps to set an `ImageBitmapRenderingContext`'s output bitmap.

```
a_canvas.probablySupportsContext("<value>"[,<arguments>]);
```

returns `true` if the canvas supports the context specified by `value`.

context=offscreenCanvas.getContext(contextId[,options]) returns an object that exposes an API for drawing on the `OffscreenCanvas` object. `contextId` may be `"2d"` or `"webgl"`. Returns `null` if the canvas has already been initialised with another context type.

offscreenCanvas=new OffscreenCanvas(width,height) returns a new `OffscreenCanvas` object that is not linked to a placeholder canvas element, and whose bitmap's size is determined by the width and height arguments.

offscreenCanvas.height[=v] set or return the height of a `"2d"` `OffscreenCanvas` object's bitmap.

offscreenCanvas.width[=v] set or return the width of a `"2d"` `OffscreenCanvas` object's bitmap.

You can give a context a new rendering context with

```
a_context=new CanvasRenderingContext2D([<width>,<height>]);
```

Control of a canvas element can be handed over to a proxy, for example, another HTML element such as an `<iframe>` element which is not able to access a canvas directly, with:

```
proxy=a_canvas.transferControlToProxy();
```

while

```
a_canvas.setContext("<context>");
proxy.setContext("<context>");
```

binds the canvas or the proxy to that context.

To reset a `<canvas>` element, simply declare its width or height, for example,

```
a_canvas.width=" ";
```

or

```
a_canvas.width=a_canvas.width;
```

but note that this does not reset the *origin-clean* flag.

15.5.2 2d context

You can save an existing canvas image as an `image/png` at 96dpi or the native density of the image (using the second option), unless the optional `<type>` argument is `image/jpeg`,[18] to a URL with:

```
<url>=a_canvas.toDataURL([<type>]);
<url>=a_canvas.toDataURLHD([<type>]);
```

or to a Javascript object which can be accessed using the `<callback>` with:

```
a_canvas.toBlob(<callback>[,<type>]);
a_canvas.toBlobHD(<callback>[,<type>]);
```

You can find the resolution at which a bitmap will be drawn with

```
<window>.screen.canvasResolution;
```

The Javascript properties and methods relating to the 2d rendering context are set out in *Some notes on Javascript*.

15.5.3 WebGL context

WebGL is a graphical rendering API on feature parity with OpenGL 2.0 though each has a few features not supported by the other.

It provides a 3D rendering pipeline for the `<canvas>` element in which you provide the vertex data, the index lists and the textures; it then updates the frame buffer, with shaders rendered using the GPU.

Each state represents an object; animation is achieved with a sequence of states in which there are timed changes in a vertex.

[18] If saving to a JPEG, a number between 0.0 and 1.0 may be added as an argument to specify the compression.

16 Microdata

16.1 Introduction

> Microdata annotates the DOM with scoped name/value pairs from custom vocabularies (Pilgrim, 2010, p. 164).

The values are existing values within the HTML code; the properties (names) may be *ad hoc* properties defined only in the current page or properties defined in publicly available machine readable vocabularies such as `The hcard vocabulary`. In database terms, microdata allows you to associate a value within the website with a field in a database table.

Currently only some search engines use the publicly available vocabularies, the advantage to the user being that the annotations enable the search engine to extract the most relevant values from the site for display in a response to a query.

You can only associate an HTML element with one vocabulary/database. So care needs to be taken in selecting the elements within which to make associations.

Each name/value pair is called an `item`; the value of each `item` is derived from the first of the following to occur:

- the value of the `itemscope` attribute
- the `content` attribute of a `<meta>` element
- the `src` attribute of an `<audio>`, `<embed>`, `<iframe>`, `<img>`, `<source>`, `< track>` or `<video>` element
- the `href` attribute of an `<a>`, `< area>` or `<link>` element
- the `data` attribute of an `<object>` element
- the `value` attribute of a `< data>` or `< meter>` element
- the `datetime` attribute of a `< time>` element
- the text of the element

where the vocabulary is confined to a single page. To declare a publicly available vocabulary, add the attribute `itemtype=" "` which takes a URL pointing to a vocabulary as its value, for example,

```
itemscope itemtype="http://microformats.org/profile/hcard"
```

Thereafter individual values within an element can be associated with a property/field by adding the attribute:

```
itemprop=" "
```

where `itemprop` may take an *ad hoc* property or any of the properties listed in a publicly available table such as those in sections 16.3 and 16.4.

For example, in

```
<div itemscope>
```

```
    John was born on <time itemprop="birthday" datetime="2009-05-10">May 10th 2009</time>.

</div>
```

the *ad hoc* property `birthday` takes the value `datetime="2009-05-10"` whereas in

```
<article itemscope itemtype="http://microformats.org/profile/hcard">
<h1 itemprop="fn">Linus Torvalds</h1>
```

the property defined as `fn` in "`http://microformats.org/profile/hcard`" takes the value ‘Linus Torvalds.’

Where the value of an `itemprop` property is only part of a string or two properties are combined in a single string, for example, Chief Software Engineer at the Linux Foundation, use `<span>` to distinguish them, for example,

```
<p><span itemprop="title">Chief Software Engineer</span> at the
<span itemprop="organisation-name">Linux Foundation</span></p>
```

In order to create sub-properties of a declared property, a further `itemscope` declaration may be contained within an `itemprop` assignment; for example,

```
<div itemscope>

    <p>Game: <span itemprop="name">Rugby</span></p>
    <p>Code: <span itemprop="code" itemscope> <span itemprop="name">Union</span>
    (<span itemprop="size">15</span> players)</span></p>

</div>
```

To associate *ad hoc* name/value pairs, an `itemscope` declaration may be given an `id=" "` attribute and an `itemref=" "` attribute containing the values of the `id=" "` attributes of elements containing `itemprop` associations, for example,

```
<div itemscope id="Sport" itemref="a b"></div>
<p id="a">Game: <span itemprop="name">Rugby</span></p>
<div id="b" itemprop="code" itemscope itemref="c"></div>
<div id="c">

    <p>Code: <span itemprop="name">Union</span></p>
    <p>Team size: <span itemprop="size">15</span> players</p>

</div>
```

Note that multiple properties may be associated with a single value, for example,

```
<div itemscope>
```

```
    <span itemprop="favourite-colour favourite-fruit">orange</span>
</div>
```

as well as multiple values with a single property.

It is also possible to use global identifiers declared within a vocabulary/table declared with `itemtype=" "` by adding the `itemid=" "` attribute which takes a global identifier such as an ISBN.

Note that microdata in no way changes the content of a document.

16.2 Microdata attributes

Microdata must have the attribute:

- `itemscope` a Boolean attribute which declares an `item` which may have the attribute `itemtype=" "` whose value is a URL pointing to a vocabulary, in which case it may have the attribute `itemid=" "` whose value is a global identifier within the vocabulary

and may have the attributes:

- `itemref=" "` whose, space separated, values are `id=" "` attributes of elements
- `itemprop=" "` associating a value with a property in a vocabulary pointed to by the `itemtype=" "` attribute or with a property in an element.

16.3 The vcard vocabulary (http://microformats.org/profile/hcard)

Property	Sub-property	Description
kind		May be `individual`, `group`, `org`, `location`
fn		Full name
n		Name (consisting of zero or more of the following sub-properties)
	family-name	Any number of family names of a person or the full name of an organisation
	given-name	Any number of given names of a person
	additional-name	Any additional names of a person
	honorific-prefix	Any honorific prefixes
	honorific-suffix	Any honorific suffixes
nickname		Any nicknames of the person or organisation
photo		A link to an image (any number are permitted)
bday		A valid date string giving the birthday of the person or organisation
anniversary		A valid date string giving the birthday of the person or organisation

Property	Sub-property	Description
sex		The biological sex of a person; may be F, M, N(/A), O(ther) or U(nknown)
gender-identity		The gender identity of a person
adr		The delivery address of a person or organisation (consisting of zero or more of the following sub-properties)
	type	May take one of the values `home` or `work` (one `adr` field within a document may also have the value of the `type` field as `pref`)
	post-office-box	A post office box (any number permitted)
	extended-address	An additional value to the address (any number permitted)
	street-address	A street address (any number permitted)
	locality	(optional) A locality (only one per `adr` field)
	region	(optional) A state o province (only one per `adr` field)
	postal-code	(optional) A postal code (only one per `adr` field)
	country-name	(optional) A country name (only one per `adr` field)
tel		The telephone number of a person or organisation (any number permitted)
	type	May take any number of the values `home`, `work`, `text voice`, `fax`, `cell`, `video`, `pager`, `textphone` (one `tel` field within a document may also have the value of the `type` field as `pref`)
	value	Contains the formatted number corresponding to a telephone number
email		An email address (any number permitted)
impp		The URL fo an instant messaging service (any number permitted)
lang		The BCP47 languages understood by the person or organisation
tz		A time zone for the person or organisation with the format `±`nn`:`nn
geo		A geographic location in LATLON format `±`$n.n$`;±`$n.n$ decimal degrees
title		The person's job title or functional role in the organisation (for example, Financial Manager) (any number permitted)
role		The person's occupation or business category (for example, Accountant) (any number permitted)
logo		A link to the logo of a person or organisation (any number permitted)
agent		The details of an agent who will act on behalf of the person or organisation (any number permitted)
org		The organisation's name and units as text string or contained in the following properties (any number permitted)
	organisation-name	The name of the organisation (only one permitted)

Property	Sub-property	Description
	organisation-unit	The unit of the organisation (any number permitted)
member		A link to a member of the group (any number permitted)
related		A relationship to another entity
	url	A link to that entity
	rel	May take one of the values `emergency`, `agent` (`contact`, `acquaintance`, `friend`, `met`, `worker`, `colleague`, `resident`, `neighbor`, `child`, `parent`, `sibling`, `spouse`, `kin`, `muse`, `crush`, `date`, `sweetheart`, `me`)
categories		Any number of categories
note		Any number of notes about a person or organisation
rev		The revision date in datetime format (any number permitted)
sound		A link to a sound for the person o organisation (any number permitted)
uid		A unique identifier for the person or organisation
url		A URL for the person or organisation (any number permitted)

16.4 The vevent vocabulary (http://microformats.org/profile/hcalendar#vevent)

Property	Description
attach	A link to a URL for the event (any number permitted)
categories	Any number of categories
class	May take one of the values `public`, `private` or `confidential` but they are purely advisory
comment	A comment about the event (any number permitted)
description	A description of the event
geo	A geographic location in LATLON format ±$n.n$;±$n.n$ decimal degrees
location	The location of the event
resources	A resource needed for the event (any number permitted)
status	May be `tentative`, `confirmed` or `cancelled`
summary	A short summary of the event
dtend	The ending date and time in datetime format.
dtstart	The starting date and time of the event in datetime format.
duration	The duration of the event (only if there is no `dtend` property)

PROPERTY	DESCRIPTION
transp	May take the values `opaque` or `transparent` indicating whether the event should be considered as taking time in a calendar
contact	Contact information for the event (any number permitted)
url	A link to the event details page.
uid	A unique identifier for the event
exdate	A datetime when a recurring event will not take place (any number permitted)
exrule	A rule for determining when an event does not occur
rdate	A date in datetime format when the event recurs (any number permitted)
rrule	A rule for determining when an event occurs
created	The date in datetime format when the record was created
last-modified	The date in datetime format when the record was last moified
sequence	The revision number for the record of the event

Table 4: The licence vocabulary (http://n.whatwg.org/work)

PROPERTY	DESCRIPTION
work	The work being described. The value must be an absolute URL.
title	The name of the work.
author	The name or contact information of one of the authors or creators of the work. May be an item with the type http://microformats.org/profile/hcard or text.
license	One of the licenses under which the work is available. The value must be an absolute URL.

17 Comments

A comment may be inserted with

```
<!-- this is a comment -->
```

18 Notes on attributes

18.1 Global attributes

The following global attributes may be specified on any HTML element:

- `accesskey=" "` specifies one or more space separated keyboard shortcuts, each of which may not be more than one Unicode point in length and which enable the user to select an option without using a pointing device; it should have an associated `title=" "` attribute saying what the keyboard shortcut gives access to.
- `autocapitalize=" "` may be used in a <`textarea`> or <`input`> element but not an `<input>` element whose `type` is `url`, `password` or `email` and may take one of the values:
  - `off` or `none` all characters to be lowercase
  - `on` or `sentences` the first character of a sentence to be uppercase
  - `words` the first character of a word to be uppercase
  - `characters` all characters to be uppercase

  In its absence, the user agent decides whether to autocapitalize.
- `autofocus` see autofocus on page 39; it may apply to all elements, not just form controls.
- `class=" "` may contain multiple values separated by spaces; these classifications should relate to the content of the element rather than its presentation; elements with a `class=" "` attribute may be selected in a CSS style sheet using the `e.value` construct (where `value` matches a value in a `class=" "` attribute).
- `contenteditable=" "` specifies that the content is editable; it takes the attributes `true` (`the` default unless the next parent element with the attribute set has the attribute `false`), `plaintext-only` or `false`.
- `dir=" "` takes the values
  - `ltr` left to right text
  - `rtl` right to left text
  - `auto` infers the direction from the first character of the text.

  It should be used explicitly rather than in CSS; the following attributes will respect its value:
  - `abbr` on <`th`> elements;
  - `alt` on <`area`>, <`img`> and <`input`> elements;
  - `content` on <`meta`> elements, if the name attribute specifies a metadata name whose value is primarily intended to be human-readable rather than machine-readable;
  - `label` on <`menu`>, <`optgroup`>, <`option`> and <`track`> elements;
  - `placeholder` on <`input`> and <`textarea`> elements;
  - `title` on all HTML elements.

  In its absence an <`input`> element whose `type="Telephone"` is assumed to be `ltr`.

  If it is set to `auto` with a <`textarea`> element or an <`input`> element whose `type="E-mail"`, `"Search"`, `"Text"`, `"Telephone"` or `"URL"` direction is inferred from the first character of its value or from its parent element's directionality.
- `disabled` a Boolean attribute indicating that the element is not enabled at present but might be at another time; effectively, it applies only to the <`button`>, <`input`>, <`select`>, <`textarea`> and <`option`> elements and the <`optgroup`> and <`fieldset`> elements whose disabled attribute is set; the property `aria-disabled` is `true` if `disabled` is `true`
- `draggable=" "` specifies that the content of the element is draggable; it takes the attributes `true`, `false` or `auto` (the default);

- `enterkeyhint` may be used in forms or contenteditable elements; it may take one of the following values:
  - `enter` present a cue for the operation 'enter,' typically inserting a new line.
  - `done` present a cue for the operation 'done,' typically meaning there is nothing more to input.
  - `go` present a cue for the operation 'go,' typically taking the user to the text they typed.
  - `next` present a cue for the operation 'next,' typically taking the user to the next field that will accept text.
  - `previous` present a cue for the operation 'previous,' typically taking the user to the previous field that will accept text.
  - `search` present a cue for the operation 'search,' typically taking the user to the results of searching for the text they have typed.
  - `send` present a cue for the operation 'send,' typically delivering the text to its target.
- `hidden` a Boolean attribute specifying that the content or input is not to be provided for or by the user; the property `aria-hidden` is `true` if `hidden` is `true`; other elements should not point to an element with the `hidden` attribute set.
- `id=" "` specifies a unique identifier (ensure it is unique!); it must contain at least one character and may not contain any space characters.
- `inert` a Boolean attribute specifying whether the element is inert
- `inputmode=" "` may take one of the following values:
  - `none` disable virtual keyboard
  - `text` display virtual keyboard
  - `tel` display virtual keyboard capable of telephone number input
  - `url` display virtual keyboard capable of URL input
  - `email` display virtual keyboard capable of email address input
  - `numeric` display virtual keyboard capable of numeric, e.g. PIN, input
  - `decimal` display virtual keyboard capable of fractional numeric input
  - `search` display virtual keyboard optimised for search.
- `is=" "` creates a customised built-in element.
- `itemid` see Microdata on page 52.
- `itemprop` see Microdata on page 52.
- `itemref` see Microdata on page 52.
- `itemscope` see Microdata on page 52.
- `itemtype` see Microdata on page 52.
- `lang=" "` specifies the language of an element;[19] if this is not specified, the `lang` attribute of the nearest parent element that has one is assumed to be the language; if no language is specified anywhere in the page, the browser should fallback on information derived from HTTP headers.
- `nonce` cryptographic nonce used in Content Security Policy checks.
- `popover=" "` the element won't be rendered until it becomes shown, at which point it will be rendered on top of other page content; it takes the attributes:
  - `auto` to close other popovers when it is opened
  - `manual` to leave other popovers open
- `slot` assigns the <slot> element whose name attribute's value matches that slot attribute's value — but only if that <slot>

[19] For the full list of language tags, see http://www.iana.org/assignments/language-subtag-registry

element finds itself in the shadow tree whose root's host has the corresponding slot attribute value.

- `spellcheck=" "` specifies that the content of an `<input>` or `<textarea>` element or the content being handled by an editing host will be checked for spelling and grammar; it takes the attributes `true` (the default unless the next parent element with the attribute set has the attribute `false`) or `false`; n.b. a user may override the setting.
- `style=" "` takes arguments in CSS Format:

  ```
  style="color: #090; line-height: 1.2"
  ```

  for application to the element; the use of the `style=" "` attributes is deprecated other than for rapid prototyping prior to removal to a stylesheet (see style).
- `tabindex=` in an `<a>` or `<link>` element with an `href=" "` attribute, a `<button>` element, an `<input>` element whose `type` attribute is not hidden, a `<select>` or `<textarea>` element or a `<summary>` element which is the first child of a `<details>` element and in elements with the `draggable` attribute set, takes an integer to indicate the order in which the browser should focus elements for tab navigation or dragging.
- `title=" "` other than in an `<abbr>`, `<dfn>`, `<input>` or `<link>`, where it normally carries additional semantics, or in the presence of an `accesskey=" "` attribute, contains a hint, comment or short footnote;[20] if its value is empty and the element is not a `<style>` or `<dfn>` element, the `title` attribute of the nearest parent element that has one is assumed to be the title; NB: use of this attribute is discouraged because it requires a cursor to hover over the attribute to reveal its value but touch only devices such as smartphones do not permit this.
- `translate=` may take the values `yes` (an empty string takes the value `yes`), in which case the the content of all text nodes in the DOM and of the following attributes:
  – `abbr` on <th> elements
  – `alt` on `<area>`, `<img>` and `<input>` elements
  – `content` on `<meta>` elements, if the name attribute specifies a metadata name whose value is known to be translatable
  – `download` on `<a>` and `<area>` elements
  – `label` on `<menu>`, `<option>` and `<track>` elements
  – `lang` on HTML elements; must be 'translated' to match the language used in the translation
  – `placeholder` on `<input>` and `<textarea>` elements
  – `srcdoc` on `<iframe>` elements; must be parsed and recursively processed
  – `style` on HTML elements; must be parsed and recursively processed (e.g. for the values of 'content' properties)
  – `title` on all HTML elements
  – `value` on `<input>` elements whose `type="Button"` or `"Reset Button"`

  is translated into the locale language, or `no`; `translate=no` should be used with content such as names and computer programs which should not be translated into the locale language; in the absence of a string, its value is inherited from the nearest parent that has the attribute set.

18.2 ARIA attributes

18.2.1 Attributes which may be used within HTML content.

alert A message with important, and usually time-sensitive, information. See related `alertdialog` and `status`.

alertdialog A type of dialog that contains an alert message, where initial focus goes to an element within the dialog. See related

[20] A longer footnote should be inserted using the `<a href=" ">` construct.

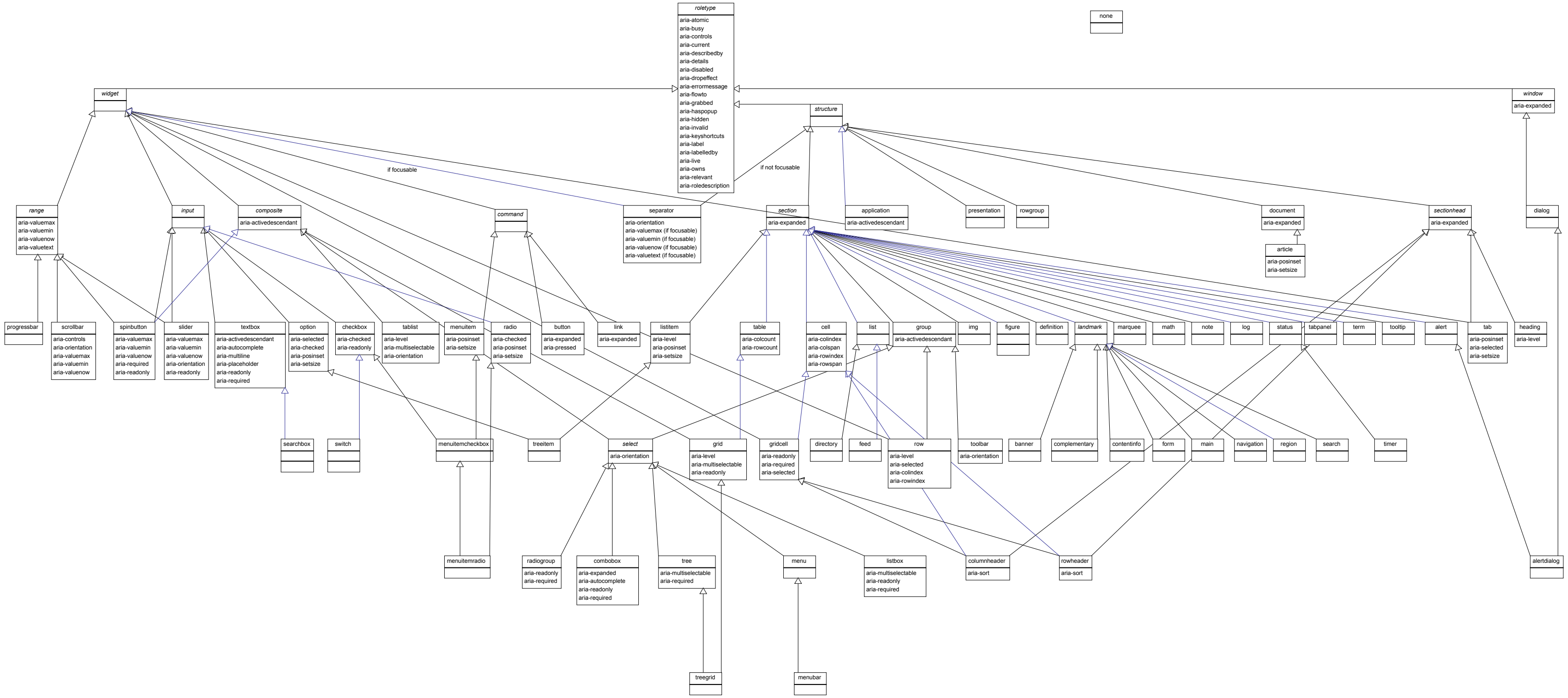


Figure 3: The RDF model of ARIA attributes

`alert` and `dialog`. Use `aria-labelledby=" "` if a visible label is present or `aria-label=" "` if not.

application A structure containing one or more focusable elements requiring user input, such as keyboard or gesture events. Use `aria-labelledby=" "` if a visible label is present or `aria-label=" "` if not.

article A section of a page that consists of a composition that forms an independent part of a document, page, or site. Where there are multiple elements with the `article` attribute, use `aria-labelledby=" "` if a visible label is present or `aria-label=" "` if not to distinguish them.

banner A region that contains mostly site-oriented content, rather than page-specific content.

blockquote A section of content that is quoted from another source.

button An input that allows for user-triggered actions when clicked or pressed. See related `link`. Only used with the `<button>` element in the unlikely case that the content of the element does not make it clear that it is a `<button>` element.

caption Visible content that names, and may also describe, a `figure`, `table`, `grid`, or `treegrid`. The `caption` must be a direct and first child of a `figure`, `table`, `grid`, or `treegrid` or the direct and last child of a `figure`. Authors SHOULD set `aria-labelledby` on the parent `figure`, `table`, `grid`, or `treegrid` to reference the element with role `caption`. However, if a `caption` contains content that serves as both a name and description for its parent, authors MAY instead set `aria-labelledby` to reference an element within the `caption` that contains a concise name, and set `aria-describedby` to reference an element within the `caption` that contains the descriptive content.

cell A cell in a tabular container. See related `gridcell`. Only used in the unlikely case that the content of the element does not make it clear what its function is. Elements with role `cell` must be contained in, or owned by, an element with the role `row`.

checkbox A checkable input that has three possible values: `true`, `false`, or `mixed`. Only used in the unlikely case that the content of the element does not make it clear what its function is.

code A section whose content represents a fragment of computer code.

columnheader A cell containing header information for a column. Only used in the unlikely case that the content of the element does not make it clear what its function is. Elements with role `columnheader` must be contained in, or owned by, an element with the role `row`.

combobox A composite widget containing a single-line `textbox` and another element, such as a `listbox` or `grid`, that can dynamically pop up to help the user set the value of the `textbox`. See related `listbox`. Requires a <label> element to be present if used with a <select> or <input> element; otherwise, use `aria-labelledby=" "` if a visible label is present or `aria-label=" "` if not.

complementary A supporting section of the document, designed to be complementary to the main content at a similar level in the DOM hierarchy, but remains meaningful when separated from the main content. Additionally, use `aria-labelledby=" "` if a visible label is present or `aria-label=" "` if not.

contentinfo A large perceivable region that contains information about the parent document. Where there are multiple elements with the `contentinfo` attribute, use `aria-labelledby=" "` if a visible label is present or `aria-label=" "` if not to distinguish them.

definition A definition of a term or concept. Use `aria-labelledby=" "` to reference the term.

deletion A deletion contains content that is marked as removed or content that is being suggested for removal. See related `insertion`.

dialog A dialog is a descendant window of the primary window of a web application, normally used to prompt the user to enter or respond to information. See related `alertdialog`. Additionally, use `aria-labelledby=" "` if a visible label is present or `aria-label=" "` if not.

directory [now deprecated]

document An element containing content that is a focusable child element of a widget or application that assistive technology users may want to browse in a reading mode. Only used in the unlikely case that the relevant widget or application does not make it clear what its function is.

emphasis One or more emphasized characters. Use only if its absence would change the meaning of the content. See related `strong`.

feed A scrollable list of articles where scrolling may cause articles to be added to or removed from either end of the list. If the nature of the feed is unclear, use `aria-labelledby=" "` if a visible label is present or `aria-label=" "` if not.

figure A perceivable section of content that typically contains a graphical document, images, code snippets, or example text. Provide a reference to the `figure` from the main text or reference descriptive content serving as a `caption` using `aria-describedby` or reference a concise name serving as a label using `aria-labelledby` or provide a label using `aria-label` in the absence of a `<figcaption>` element

form A landmark region that contains a collection of items and objects that, as a whole, combine to create a form. See related `search`. If the purpose of the `<form>` element is unclear, use `aria-labelledby=" "` if a visible label is present or `aria-label=" "` if not.

generic A nameless container element that has no semantic meaning on its own.

grid A collection of one or more rows with one or more cells where some or all cells in the grid are focusable by using methods of two-dimensional navigation, such as directional arrow keys. A `grid` may only contain elements with role `gridcell`, `columnheader`, or `rowheader` owned by elements with role `row`, which are in turn owned by an element with role `rowgroup`, or `grid`. In the absence of a `<caption>` element in a `<table>` element, use `aria-labelledby=" "` if a visible label is present or `aria-label=" "` if not.

gridcell A cell in a grid that may be focusable, editable, and selectable. Only used in the unlikely case that the content of the element does not make it clear what its function is.

group A set of user interface objects which are not intended to be included in a page summary or table of contents by assistive technologies. Opposite of a `region`. Do not use with `<fieldset>` element with a `<legend>` element, a `<details>` element with a `<summary>` element or an `<optgroup>` element with a `<label>` element; otherwise, use `aria-labelledby=" "` if a visible label is present or `aria-label=" "` if not.

heading A heading for a section of the page (if not an `<6.6>` element).

img A container for a collection of elements that form an image. For the `<img>` element, use the `alt=" "` attribute and, for other elements with the `img` role, use `aria-labelledby=" "` if a visible label is present or `aria-label=" "` if not.

insertion An insertion contains content that is marked as added or content that is being suggested for addition. See related `deletion`.

link An interactive reference to an internal or external resource that, when activated, causes the user agent to navigate to that resource. See related `button`. Only used in the unlikely case that the content of the element does not make it clear what its function is.

list A section containing `listitem` elements. See related `listbox`. Where there are multiple elements with the `list` attribute, use `aria-labelledby=" "` if a visible label is present or `aria-label=" "` if not to distinguish them.

listbox A widget that allows the user to select one or more items from a list of choices. Items within the list are static and, unlike standard HTML `<select>` elements, may contain images. List boxes contain children whose role is `option`. See related `combobox` and `list`. In the absence of a `<label>` element within a `<select>` element having either the `multiple` attribute or the `sizes` attribute greater than 1, use `aria-labelledby=" "` if a visible label is present or `aria-label=" "` if not.

listitem A single item in a list or directory whose parent has the role `list` or `group`. Not needed at present.

log A type of live region where new information is added in meaningful order and old information may disappear. See related `marquee`. Use `aria-labelledby=" "` if a visible label is present or `aria-label=" "` if not.

main The main content of a document. It offers a non-obtrusive alternative to 'skip to main content' links. If the content of the element changes, use `aria-labelledby=" "` if a visible label is present or `aria-label=" "` if not.

marquee A type of live region where non-essential information changes frequently. See related `log`. Use `aria-labelledby=" "` if a visible label is present or `aria-label=" "` if not.

math Content that represents a mathematical expression. If the element is navigable, use `aria-labelledby=" "` if a visible label is present or `aria-label=" "` if not; if the element is not navigable, use `aria-label=" "` to hold either the content or the name of the expression.

meter An element that represents a scalar measurement within a known range, or a fractional value. See related `progressbar`.

menu A type of widget that offers a list of choices to the user. Elements with the role `menu` have an implicit `aria-orientation` value of `vertical`. Use `aria-labelledby=" "` if a visible label is present or `aria-label=" "` if not.

menubar A presentation of `menu` that usually remains visible and is usually presented horizontally. A menu bar is used to create a consistent set of frequently used commands similar to the typical menu bar interaction in a desktop graphical user interface. Use `aria-labelledby=" "` if a visible label is present or `aria-label=" "` if not.

menuitem An option in a set of choices contained by a `menu` or `menubar`. Only used in the unlikely case that the content of the element does not make it clear what its function is.

menuitemcheckbox A `menuitem` with a checkable state whose possible values are `true`, `false`, or `mixed`. Only used in the unlikely case that the content of the element does not make it clear what its function is.

menuitemradio A checkable `menuitem` in a set of elements with the same role, only one of which can be checked at a time. Only used in the unlikely case that the content of the element does not make it clear what its function is.

navigation A collection of navigational elements (usually links) for navigating the document or related documents. Use `aria-labelledby=" "` if a visible label is present or `aria-label=" "` if not.

none A synonym for `presentation`.

note A section whose content is parenthetic or ancillary to the main content of the resource. Use `aria-labelledby=" "` if a visible label is present or `aria-label=" "` if not.

option A selectable item in a select list contained in, or owned by, an element with the role `listbox`. Only used in the unlikely case that the content of the element does not make it clear what its function is.

paragraph A paragraph of content.

presentation An element whose implicit native role semantics will not be mapped to the accessibility API.

progressbar An element that displays the progress status for tasks that take a long time. In the absence of a `<label>` element, use `aria-labelledby=" "` if a visible label is present or `aria-label=" "` if not.

progressbar An element that displays the progress status for tasks that take a long time.

radio A checkable input in a group of radio roles, only one of which can be checked at a time. Only used in the unlikely case that the content of the element does not make it clear what its function is.

radiogroup A group of radio buttons in a select list that can only have a single entry checked at any one time. Use `aria-labelledby=" "` if a visible label is present or `aria-label=" "` if not.

region A large perceivable section of a web page or document, that is important enough to be included in a page summary or table of contents, for example, an area of the page containing live sporting event statistics. It should not be used where that roles `main`, `complementary`, or `navigation` are more appropriate. Use `aria-labelledby=" "` if a visible label is present or `aria-label=" "` if not.

row A row of cells in a tabular container containing `cell` or `gridcell` elements. Only used if the content of the element does not make it clear what its function is and it is a descendant of a focusable element with the attribute `treegrid`.

rowgroup A structure containing one or more row elements in a tabular container. Do not use.

rowheader A cell containing header information for a row in a grid contained in, or owned by, an element with the role `grid`. Only used in the unlikely case that the content of the element does not make it clear what its function is. Ensure elements with role `rowheader` are contained in, or owned by, an element with the role `row`.

scrollbar A graphical object that controls the scrolling of content within a viewing area, regardless of whether the content is fully displayed within the viewing area. Use `aria-labelledby=" "` if a visible label is present or `aria-label=" "` if not.

search A landmark region that contains a collection of items and objects that, as a whole, combine to create a search facility. See related `form` and `searchbox`. Use `aria-labelledby=" "` if a visible label is present or `aria-label=" "` if not.

searchbox A type of `textbox` intended for specifying search criteria. See related `textbox` and `search`. In the absence of a `<label>` element, use `aria-labelledby=" "` if a visible label is present or `aria-label=" "` if not.

separator A divider that separates and distinguishes sections of content. Where there are multiple elements with the `separator` attribute, use `aria-labelledby=" "` if a visible label is present or `aria-label=" "` if not to distinguish them.

slider A user input where the user selects a value from within a given range. In the absence of a `<label>` element within an `<input>` element, use `aria-labelledby=" "` if a visible label is present or `aria-label=" "` if not.

spinbutton A form of range that expects the user to select from among discrete choices. In the absence of a `<label>` element within an `<input>` element with the `textbox` attribute, use `aria-labelledby=" "` if a visible label is present or `aria-label=" "` if not.

status A container whose content is advisory information for the user but is not important enough to justify an alert, often but not necessarily presented as a status bar. See related `alert`.

strong Content that is important, serious, or urgent. Use only if its absence would change the meaning of the content. See related emphasis.

subscript One or more subscripted characters. Use this role only if the absence of the subscript would change the meaning of the content. See related `superscript`.

superscript One or more superscripted characters. Use this role only if the absence of the superscript would change the meaning of the content. See related `subscript`.

switch A type of checkbox that represents on/off values, as opposed to checked/unchecked values. The `aria-checked` attribute indicates whether the input is `on` (true) or `off` (false). The `mixed` value is equivalent to `false` for this role. See related `checkbox`. Only used in the unlikely case that the content of the element does not make it clear what its function is.

tab A grouping label providing a mechanism for selecting the tab content that is to be rendered to the user contained in, or owned by, an element with the role `tablist`. Only used in the unlikely case that the content of the element does not make it clear what its function is.

table A section containing data arranged in rows and columns intended for tabular containers which are not interactive. See related `grid`. In the absence of a `<caption>` element in a `<table>` element, use `aria-labelledby=" "` if a visible label is present or `aria-label=" "` if not.

tablist A list of `tab` elements, which are references to `tabpanel` elements. Use `aria-labelledby=" "` if a visible label is present or `aria-label=" "` if not.

tabpanel A container for the resources associated with a `tab`, where each `tab` is contained in a `tablist`. Use `aria-labelledby=" "` to point to the tab element that controls the `tabpanel`.

term A word or phrase with a corresponding definition. See related `definition`. Do not use.

textbox Input that allows free-form text as its value. In the absence of a `<label>` element within an `<input>` or `< textarea>` element with the `textbox` attribute, use `aria-labelledby=" "` if a visible label is present or `aria-label=" "` if not.

time An element that represents a specific point in time.

timer A type of live region containing a numerical counter which indicates an amount of elapsed time from a start point, or the time remaining until an end point. Use `aria-labelledby=" "` if a visible label is present or `aria-label=" "` if not.

toolbar A collection of commonly used function buttons or controls represented in compact visual form, often a subset of functions found in a `menubar`, which have an implicit `aria-orientation` value of `horizontal`. Where there are multiple elements with the `toolbar` attribute, use `aria-labelledby=" "` if a visible label is present or `aria-label=" "` if not to distinguish them.

tooltip A contextual popup that displays a description for an element. Only used in the unlikely case that the content of the element does not make it clear what its function is.

tree A type of list that may contain sub-level nested groups that can be collapsed and expanded. Use `aria-labelledby=" "` if a visible label is present or `aria-label=" "` if not.

treegrid A grid whose rows can be expanded and collapsed in the same manner as for a `tree`. In the absence of a `<caption>` element within a `<table>` element, use `aria-labelledby=" "` if a visible label is present or `aria-label=" "` if not.

treeitem An option item of a `tree`. This is an element within a tree contained in, or owned by, an element with the role `group` or `tree` that may be expanded or collapsed. Only used in the unlikely case that the content of the element does not make it clear what its function is.

18.2.2 Abstract attributes which should not be used within HTML content

command A form of widget that performs an action but does not receive input data.

composite A widget that may contain navigable descendants or owned children.

input A generic type of widget that allows user input.

landmark A perceivable section containing content that is relevant to a specific, author-specified purpose and sufficiently important that users will likely want to be able to navigate to the section easily and to have it listed in a summary of the page.

range An input representing a range of values that can be set by the user.

roletype The base role from which all other roles in this taxonomy inherit.

section A renderable structural containment unit in a document or application.

sectionhead A structure that labels or summarises the topic of its related section.

select A form widget that allows the user to make selections from a set of choices.

structure A document structural element.

widget An interactive component of a graphical user interface (GUI).

window A browser or application window.

In addition, `aria-*=" "` attributes are set by the browser to reflect particular states of elements on a webpage, of which:

`aria-labelledby=" "` should be used for a string when there is visible text content unrelated to the string that will serve as an appropriate accessible name, and

`aria-label=" "` for a string when there is no visible text content that will serve as an appropriate accessible name.

18.3 Other notes on attributes

1. All Boolean attributes are `true`; to avoid confusion, they cannot take the attribute `"false"`; to indicate a `false` state, omit the attribute.
2. Some attributes have an *invalid value default* and a *missing value default* which may not be the same.
3. `target= " "` attributes beginning `"_"` are restricted to specific predefined values so that predefined values may be introduced in future which do not conflict with author-defined values; the following values are predefined:
   - `_blank` opens a new browsing context — as does a string naming an unknown browsing context or naming a browsing context not in the current hierarchy
   - `_self` opens in the current browsing context — as does an empty string
   - `_parent` opens in the parent browsing context, if any
   - `_top` opens in the top browsing context if the current browsing context is not already the top browsing context.
4. Other special attributes:
   - `data-*=" "` attributes are ignored by browsers; they are intended for use only by scripts within the page that created the attribute, for example, to enable users to search for content based on such attributes
   - any attribute beginning with `x-` is guaranteed never to be added to the formal HTML syntax and can therefore be used experimentally; they normally take the form `x-<vendor>-<feature>`.[21]

[21] The corresponding IDL attribute would be `vendorFeature`.

19 MIME types

application/atom+xml	Atom
application/json	JSON
application/octet-stream	generic binary data [RFC2046]
application/microdata+json	microdata as JSON
application/rss+xml	RSS
application/x-www-form-urlencoded	form submission
application/xhtml+xml	HTML
application/xml	XML [RFC7303]
image/gif	GIF images
image/jpeg	JPEG images
image/png	PNG images
image/svg+xml	SVG images
multipart/form-data	form submission [RFC7578]
multipart/mixed	generic mixed content [RFC2046]
multipart/x-mixed-replace	streaming server push
text/css	CSS
text/event-stream	server-sent event streams
text/javascript	JavaScript [RFC9239]
text/json	JSON (legacy type)
text/plain	generic plain text [RFC2046]
text/html	HTML
text/ping	hyperlink auditing
text/uri-list	list of URLs [RFC2483]
text/vcard	vCard [RFC6350]
text/vtt	WebVTT [WEBVTT]
text/xml	XML [RFC7303]
video/mp4	MPEG-4 video [RFC4337]
video/mpeg	MPEG video [RFC2046]

References

Bah, T. (2011). *Inkscape: guide to a vector drawing program* (Fourth ed.). London: Prentice Hall.

Fisher, D. (2010, 25 August). HTML5. Talk at the Bradford GNU/LUG meeting at Bradford Council for Voluntary Services, Bradford, West Yorkshire.

Flanagan, D. (2011). *JavaScript: the definitive guide* (Sixth ed.). Sebastapol, CA: O'Reilly Media.

Hickson, I. (editor). (2011, 18 March). HTML: living standard. http://www.whatwg.org/specs/web-apps/current-work/html-a4.pdf.

Hickson, I. (editor). (2013, 12 July). HTML: living standard. `http://www.whatwg.org/specs/web-apps/current-work/`.

Hickson, I. (editor). (2014, 6 August). HTML: living standard. `http://www.whatwg.org/specs/web-apps/current-work/`.

Hickson, I. (editor). (2015, 21 July). HTML: living standard. `https://html.spec.whatwg.org/`.

Hickson, I. (editor). (2018, 19 February). HTML: living standard. `https://html.spec.whatwg.org/`.

Hickson, I. (editor). (2020, 30 March). HTML: living standard. `https://html.spec.whatwg.org/`.

Hickson, I. (editor). (2023, 11 September). HTML: living standard. `https://html.spec.whatwg.org/`.

Pilgrim, M. (2010). *HTML5: up and running*. Sebastopol CA: O'Reilly Media.

A HTML APIs

The following APIs are available:

Geolocation a scripting API that uses World Geodetic System coordinates. The API enables Javascript scripts to perform a variety of actions once the user has given consent through the browser.

Drag and Drop allows elements who attribute has been set to `draggable` to be moved.

Web storage allows data to be stored locally or just for one session.

Web workers are scripts which run in the background rather than holding up current operations.

Server sent events allow browsers to receive update messages from servers.

A summary of the features of the APIs is in *Some notes on Javascript*.

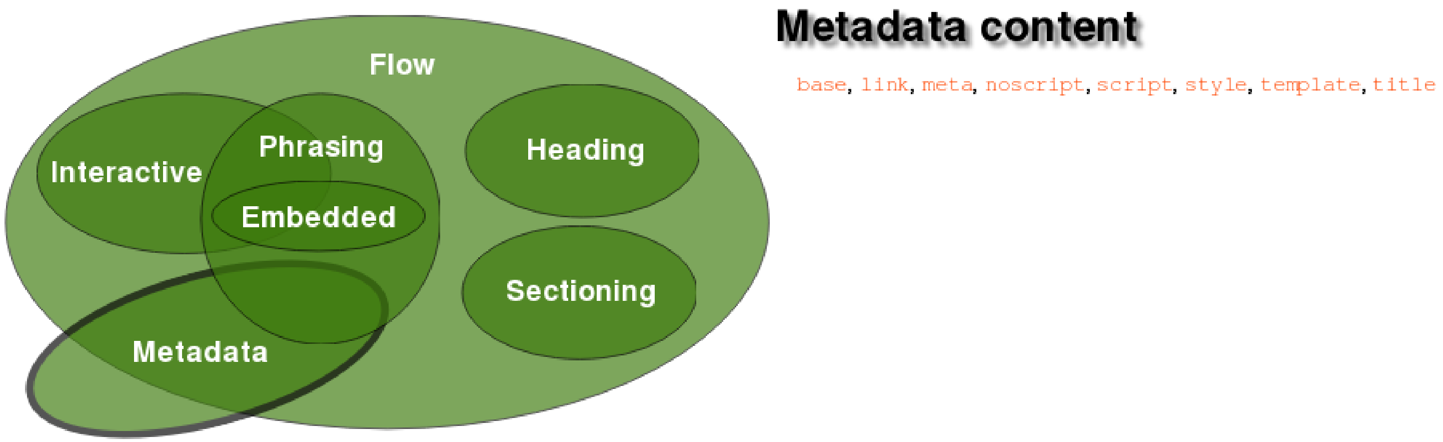


Figure 4: Metadata content (Hickson, 2013)

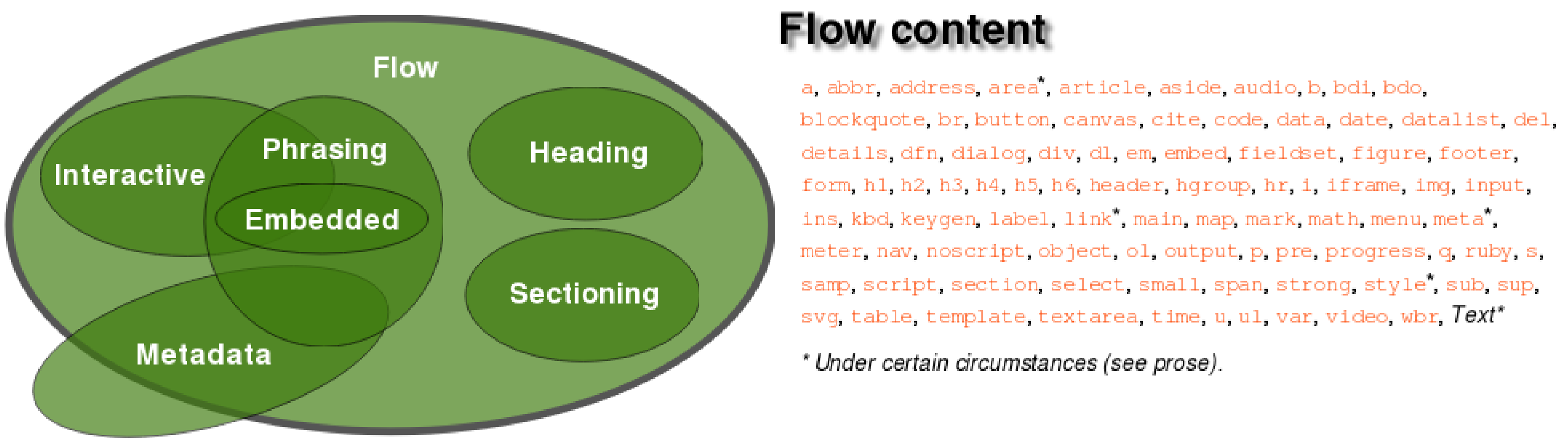
B HTML elements by groups

B.1 Metadata elements

- base
- link
- meta
- noscript
- script
- style
- template
- title

B.2 Flow elements

- a
- abbr
- address
- area (if it is a descendant of a <map> element)
- article
- aside
- audio
- b
- bdi (see page 24)
- bdo (see page 24)
- blockquote
- br
- button
- canvas
- cite
- code
- data
- datalist
- del (see page 25)
- details
- dfn
- dialog
- div
- dl (see page 18)
- em
- embed
- fieldset



Flow content

a, abbr, address, area*, article, aside, audio, b, bdi, bdo, blockquote, br, button, canvas, cite, code, data, date, datalist, del, details, dfn, dialog, div, dl, em, embed, fieldset, figure, footer, form, h1, h2, h3, h4, h5, h6, header, hgroup, hr, i, iframe, img, input, ins, kbd, keygen, label, link*, main, map, mark, math, menu, meta*, meter, nav, noscript, object, ol, output, p, pre, progress, q, ruby, s, samp, script, section, select, small, span, strong, style*, sub, sup, svg, table, template, textarea, time, u, ul, var, video, wbr, *Text**

** Under certain circumstances (see prose).*

Figure 5: Flow content (Hickson, 2013)

figure

footer

form

h1 (see page 14)

h2 (see page 14)

h3 (see page 14)

h4 (see page 14)

h5 (see page 14)

h6 (see page 14)

header

hgroup

hr

i

iframe

img

input

ins (see page 25)

kbd

label

link (if the `itemprop` attribute (see Microdata on page 52) is present)

`main`

map

mark

math

menu

meta (if the `itemprop` attribute is present)

meter

nav

noscript

object

ol

output

p

picture

pre

progress

q

ruby (see page 23)

s

samp

script

section

select

slot

small

span

strong

sub (see page 23)

sup (see page 23)

svg

table

template

textarea

time

u

ul

var

video

wbr

text

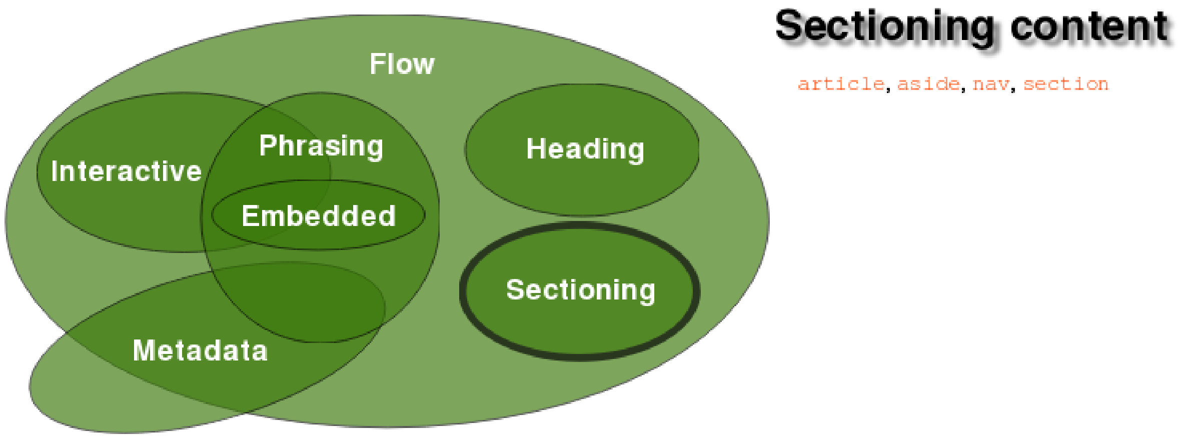


Figure 6: Sectioning content (Hickson, 2013)

B.3 Sectioning elements

article

aside

nav

section

B.4 Heading elements

h1 (see page 14)
h2 (see page 14)
h3 (see page 14)
h4 (see page 14)
h5 (see page 14)
h6 (see page 14)
hgroup

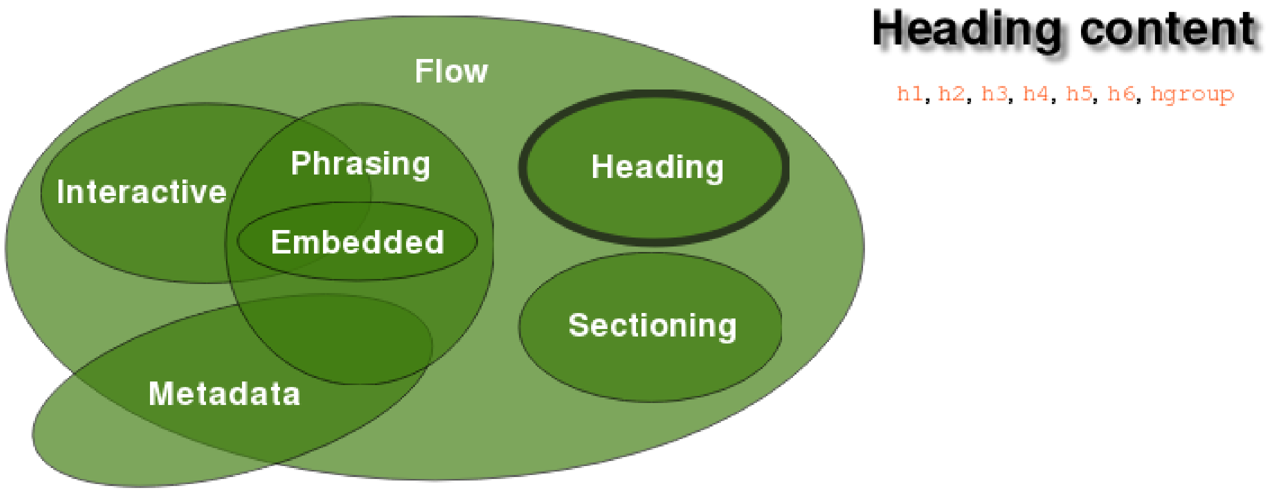


Figure 7: Heading content (Hickson, 2013)

B.5 Phrasing elements

a

abbr

area (if it is a descendant of a <map> element)

audio

b

bdi (see page 24)

bdo (see page 24)

br

button

canvas

cite

code

data

datalist

del (if it contains only phrasing content) (see page 25)

dfn

em

embed

i

iframe

img

input

ins (if it contains only phrasing content) (see page 25)

kbd

label

link (if the itemprop attribute (see Microdata on page 52) is present)

map (if it contains only phrasing content)

mark

math

meta (if the itemprop attribute is present)

meter

noscript

object

output

picture

progress

q

ruby (see page 23)

s

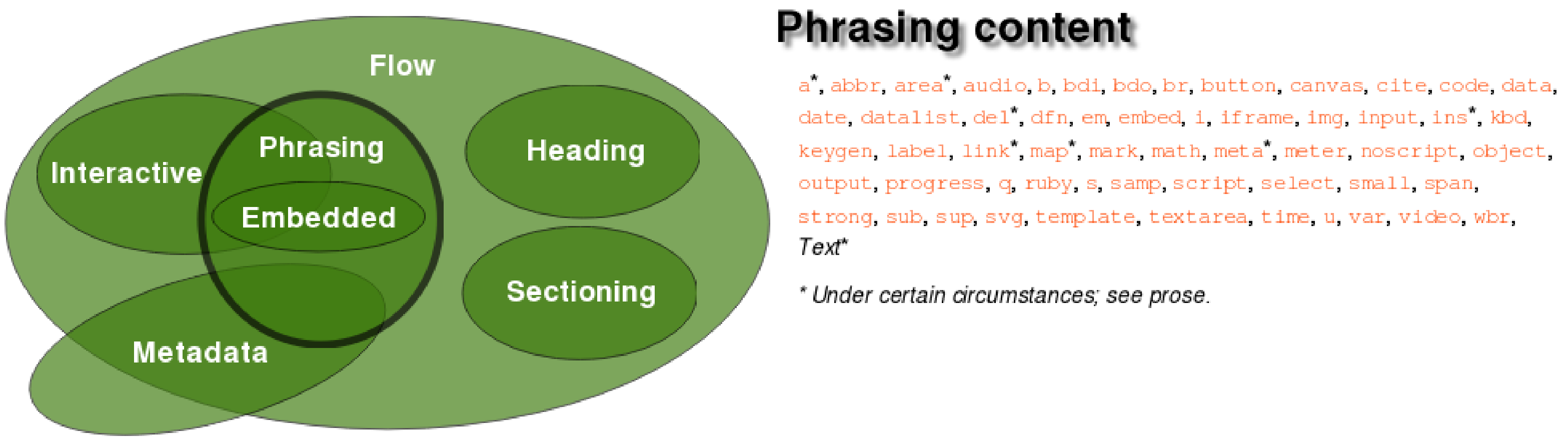
samp

script

select

slot

small



Phrasing content

a*, abbr, area*, audio, b, bdi, bdo, br, button, canvas, cite, code, data, date, datalist, del*, dfn, em, embed, i, iframe, img, input, ins*, kbd, keygen, label, link*, map*, mark, math, meta*, meter, noscript, object, output, progress, q, ruby, s, samp, script, select, small, span, strong, sub, sup, svg, template, textarea, time, u, var, video, wbr, *Text**

* *Under certain circumstances; see prose.*

Figure 8: Phrasing content (Hickson, 2013)

span
strong
sub (see page 23)
sup (see page 23)
svg
template
textarea
time
u
var
video
wbr
text

B.6 Embedding elements

audio
canvas
embed
iframe
img
math
object
picture
svg
video

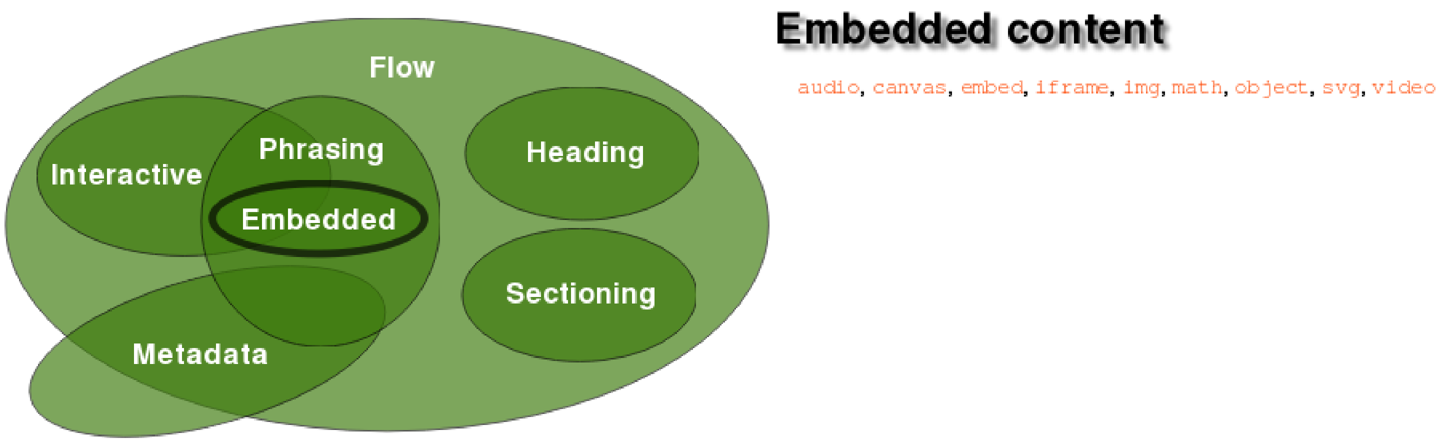


Figure 9: Embedded content (Hickson, 2013)

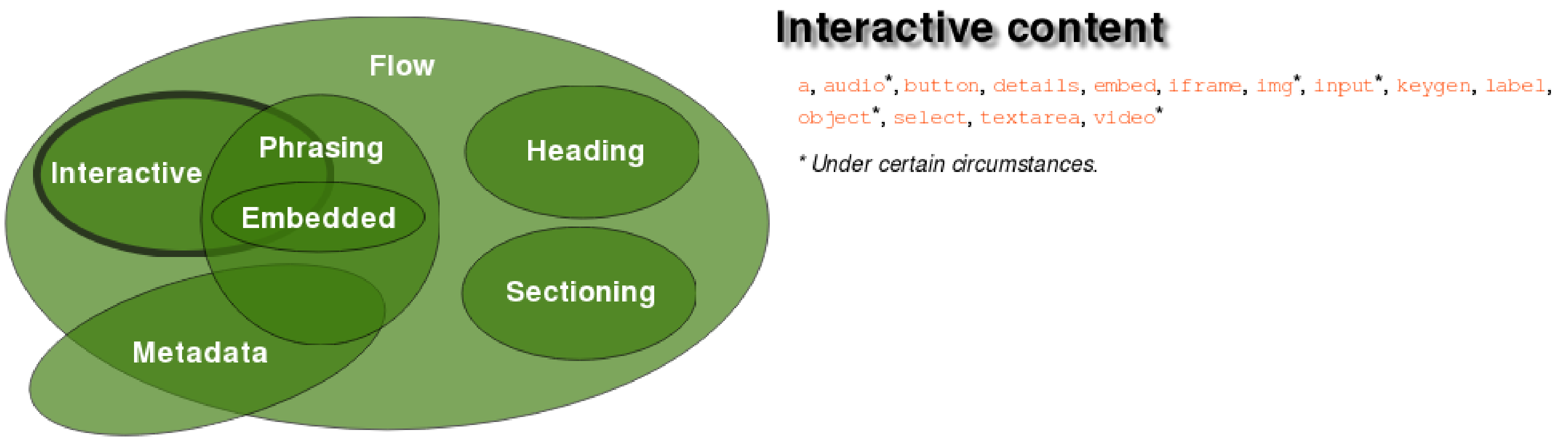
B.7 Interactive elements

a
audio (if the `controls` attribute is present)
button
details
embed
iframe
img (if the `usemap` attribute is present)
input (if the `type` attribute is *not* in the `hidden` state)
label
object (if the `usemap` attribute is present)
select
textarea
video (if the `controls` attribute is present)

The `tabindex` attribute (section 18.1 on page 58) can make any element into an interactive element.

B.8 Palpable content

a
abbr
address
article
aside
audio (if the `controls` attribute is present)
b
bdi (see page 24)
bdo (see page 24)
blockquote
button
canvas
cite
code
data



Interactive content

a, audio*, button, details, embed, iframe, img*, input*, keygen, label, object*, select, textarea, video*

* Under certain circumstances.

Figure 10: Interactive content (Hickson, 2013)

details
dfn
div
dl (see page 18; if the element's children include at least one name-value group)
em
embed
fieldset
figure
footer
form
h1 (see page 14)
h2 (see page 14)
h3 (see page 14)
h4 (see page 14)
h5 (see page 14)
h6 (see page 14)
header
hgroup
i
iframe
img
input (if the `type` attribute is *not* in the `hidden` state)
ins (see page 25)
kbd
label
main
map
mark
math
menu (if the `type` attribute is in the `toolbar` state),
meter
nav
object
ol (if the element's children include at least one `<li>` element),
output
p
pre
progress
q
ruby (see page 23)
s
samp
section
select
small
span
strong
sub (see page 23)
sup (see page 23)
svg
table
textarea
time
u
ul (if the element's children include at least one `<li>` element),
var
video
text that is not inter-element whitespace

B.9 Script supporting elements

script

template

C Example of a DOM tree for some HTML text

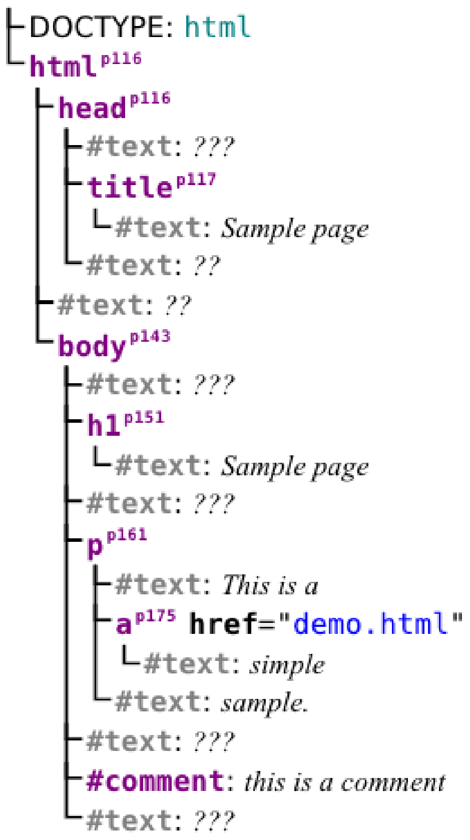
The DOM tree for this HTML text shown in Figure 11 on the following page is taken from Hickson (2011, pp. 20–21):

```
<!DOCTYPE html>
<html>

    <head>
        <title>Sample page</title>
    </head>
    <body>
        <h1>Sample page</h1>
        <p>This is a <a href="demo.html">simple</a> sample.</p>
        <!-- this is a comment -->
    </body>

</html>
```

A DOM tree has a `DocumentType`, `Element`, `Text` and `Comment` nodes and may have `ProcessingInstruction` nodes; it can be manipulated by using `<script>` elements.



```
├DOCTYPE: html
└html p116
  ├head p116
  │ ├#text: ⏎␣␣
  │ ├title p117
  │ │ └#text: Sample page
  │ └#text: ⏎␣
  ├#text: ⏎␣
  └body p143
    ├#text: ⏎␣␣
    ├h1 p151
    │ └#text: Sample page
    ├#text: ⏎␣␣
    ├p p161
    │ ├#text: This is a
    │ ├a p175 href="demo.html"
    │ │ └#text: simple
    │ └#text: sample.
    ├#text: ⏎␣␣
    ├#comment: this is a comment
    └#text: ⏎␣⏎
```

Figure 11: DOM tree

D Encoding video formats

D.1 Script for encoding video

```
## Theora/Vorbis/Ogg
you@localhost$ ffmpeg2theora --videobitrate 200 --max_size 320x240 --output pr6.ogv pr6.dv
## H.264/AAC/MP4
you@localhost$ HandBrakeCLI --preset "iPhone & iPod Touch" --vb 200 --width 320 --two-pass --turbo
--optimize --input pr6.dv --output pr6.mp4
## VP8/Vorbis/WebM
you@localhost$ ffmpeg -pass 1 -passlogfile pr6.dv -threads 16 -keyint_min 0 -g 250 -skip_threshold 0
-qmin 1 -qmax 51 -i pr6.dv -vcodec libvpx -b 204800 -s 320x240 -aspect 4:3 -an -f webm -y NUL
you@localhost$ ffmpeg -pass 2 -passlogfile pr6.dv -threads 16 -keyint_min 0 -g 250 -skip_threshold 0
-qmin 1 -qmax 51 -i pr6.dv -vcodec libvpx -b 204800 -s 320x240 -aspect 4:3 -acodec libvorbis -ac 2 -y
pr6.webm
```

E Video MIME types

These examples of using the <source> element to specify video MIME types are from Hickson (2020, 4.8.2).

H.264 Constrained baseline profile video (main and extended video compatible) level 3 and Low-Complexity AAC audio in MP4 container

```
<source src="video.mp4" type='video/mp4; codecs="avc1.42E01E, mp4a.40.2"'>
```

H.264 Extended profile video (baseline-compatible) level 3 and Low-Complexity AAC audioMP4 container

```
<source src="video.mp4" type='video/mp4; codecs="avc1.58A01E, mp4a.40.2"'>
```

H.264 Main profile video level 3 and Low-Complexity AAC audio in MP4 container

```
<source src="video.mp4" type='video/mp4; codecs="avc1.4D401E, mp4a.40.2"'>
```

H.264 ‘High’ profile video (incompatible with main, baseline, or extended profiles) level 3 and Low-Complexity AAC audio in MP4 container

```
<source src="video.mp4" type='video/mp4; codecs="avc1.64001E, mp4a.40.2"'>
```

MPEG-4 Visual Simple Profile Level 0 video and Low-Complexity AAC audio in MP4 container

```
<source src="video.mp4" type='video/mp4; codecs="mp4v.20.8, mp4a.40.2"'>
```

MPEG-4 Advanced Simple Profile Level 0 video and Low-Complexity AAC audio in MP4 container

```
<source src="video.mp4" type='video/mp4; codecs="mp4v.20.240, mp4a.40.2"'>
```

MPEG-4 Visual Simple Profile Level 0 video and AMR audio in 3GPP container

```
<source src="video.3gp" type='video/3gpp; codecs="mp4v.20.8, samr"'>
```

Theora video and Vorbis audio in Ogg container

```
<source src="video.ogv" type='video/ogg; codecs="theora, vorbis"'>
```

Theora video and Speex audio in Ogg container

```
<source src="video.ogv" type='video/ogg; codecs="theora, speex"'>
```

Vorbis audio alone in Ogg container

```
<source src="audio.ogg" type='audio/ogg; codecs=vorbis'>
```

Speex audio alone in Ogg container

```
<source src="audio.spx" type='audio/ogg; codecs=speex'>
```

FLAC audio alone in Ogg container

```
<source src="audio.oga" type='audio/ogg; codecs=flac'>
```

Dirac video and Vorbis audio in Ogg container

```
<source src="video.ogv" type='video/ogg; codecs="dirac, vorbis"'>
```

F Omitting end tags

Void elements, <area>, <base>, <br>, <col>, <embed>, <hr>, <img>, <input>, <link>, <meta>, <source>, <track> and <wbr> have no end tags.

The end tags of an <html> or a <body> element can be omitted if they are not followed by a comment.

The end tags of a <head>, <colgroup> or <caption> element may be omitted if the element is not immediately followed by a space character or a comment.

The end tags of the following elements may be omitted if they are followed by another element of the same type or are the last element within the parent element:

<li>, <optgroup>, <tr>

The end tags of the following elements may be omitted if they are followed by another element of the same type or another element (specified in parentheses) or are the last element within the parent element:

<dd> (<dt>), <rt> (<rp>), <rp> (<rt>), <option> (<optgroup>), <tbody> (<tfoot>), <td> (<th>), <th> (<td>)

The end tag of a <dt> element may be omitted it is immediately followed by a <dt> or <dd> element.

The end tag of a <p> element may be omitted if it is immediately followed by one of the following elements or it is the last element within the parent element where the parent element is not an <a>, <audio>, <del>, <ins>, <map>, <noscript> or <video> element:

<address>, <article>, <aside>, <blockquote>, <details>, <div>, <dl>, <fieldset>, <figcaption>, <figure>, <footer>, <form>, <h1--h6>, <header>, <hgroup>, <hr>, <main>, <menu>, <nav>, <ol>, <p>, <pre>, <section>, <table>, <ul>

The end tag of a <thead> element may be omitted if it is immediately followed by a <tbody> or <tfoot> element.

The end tag of a <tbody> element may be omitted if it is immediately followed by a <tbody> or <tfoot> element or if there is no more content in the parent element.

The end tag of a <tfoot> element may be omitted if there is no more content in the parent element.