

ARIA in HTML

W3C Recommendation 05 August 2025



▼ More details about this document

This version:

<https://www.w3.org/TR/2025/REC-html-aria-20250805/>

Latest published version:

<https://www.w3.org/TR/html-aria/>

Latest editor's draft:

<https://w3c.github.io/html-aria/>

History:

<https://www.w3.org/standards/history/html-aria/>

[Commit history](#)

Implementation report:

<https://w3c.github.io/html-aria/results/implementation-results.html>

Editors:

Scott O'Hara ([Microsoft](#))

Patrick H. Lauke ([TetraLogical](#))

Former editor:

Steve Faulkner ([TPGi](#)) - Until 28 April 2023

Feedback:

[GitHub w3c/html-aria](#) ([pull requests](#), [new issue](#), [open issues](#))

public-webapps@w3.org with subject line [html-aria] ... *message topic* ... ([archives](#))

Errata:

[Errata exists](#).

See also [translations](#).

Copyright © 2025 [World Wide Web Consortium](#). W3C[®] [liability](#), [trademark](#) and [permissive document license](#) rules apply.

Abstract

This specification defines the authoring rules (author conformance requirements) for the use of [Accessible Rich Internet Applications \(WAI-ARIA\) 1.2](#) and [Digital Publishing WAI-ARIA Module 1.0](#)

attributes on [[HTML](#)] elements. This specification's primary objective is to define requirements for use with conformance checking tools used by authors (i.e., web developers). These requirements will aid authors in their development of web content, including custom interfaces and widgets, which make use of ARIA to complement or extend the features of the host language [[HTML](#)].

Status of This Document

This section describes the status of this document at the time of its publication. A list of current W3C publications and the latest revision of this technical report can be found in the [W3C standards and drafts index](#).

ARIA in HTML is an [[HTML](#)] specification module. Any HTML features, conformance requirements, or terms that this specification module makes reference to, but does not explicitly define, are defined by the [HTML Standard](#).

Since this specification became a W3C Recommendation on 09 December 2021, the following substantive additions and/or corrections have been proposed:

- [23 July 2025 - Addition](#): Update the [label](#) element to allow `role` and `aria-*` attributes to be specified when the element is not associated with a labelable element.
- [23 July 2025 - Addition](#): Add the [selectedcontent](#) element and provide updated allowances for the [button](#) element when it is used in the context of a customized `select` element.
- [23 July 2025 - Correction](#): Clarify that the [html](#) element is a generic element, and that neither the `document` or `generic` roles are recommended to be used on the element.
- [23 December 2024 - Addition](#): Update the [img](#) element to allow the `math` role.
- [13 December 2024 - Addition](#): Update to include the `image` role as preferred synonym to the `img` role.
- [13 December 2024 - Addition](#): Clarify the allowance for `aria-hidden` when used with the `hidden` attribute.
- [4 October 2023 - Addition](#): Update the `button` element and `input type=button,image,reset,submit` elements to allow the `separator` role.
- [3 October 2023 - Correction](#): Update the [img](#) element allowances to be based on if the element has an accessible name or not.
- [21 August 2023 - Addition](#): Update the [address](#) and [hgroup](#) element allowances per their updated mapping to the `group` role.

- [9 July 2023 - Addition](#): Update the [aside](#) element to allow the `dpub doc-glossary` role.
- [5 July 2023 - Addition](#): Update the [button](#), [input type=button](#), [input type=image](#), [input type=reset](#), and [input type=submit](#) elements to align their allowed roles.
- [29 June 2023 - Addition](#): Update the [s](#) element allowed roles to indicate use of `role=deletion` on the element would be considered redundant.
- [31 May 2023 - Correction](#): Conditionally revise allowed `aria-*` attributes and roles on [summary](#) element.
- [31 May 2023 - Correction](#): Update [li](#) element role allowances in context to the element's ancestral relationship, or lack of, to a list element parent.
- [24 March 2023 - Addition](#): The [search](#) element has been added.
- [6 March 2023 - Addition](#): Disallow `aria-hidden=true` on the [body](#) element.
- [13 February 2023 - Addition](#): Update [figure](#) element role allowances to include `doc-example`.
- [07 November 2022 - Correction](#): Revisions to 'any role' term description.
- [14 July 2022 - Correction](#): Disallow roles and `aria-*` attributes on the [datalist](#) element.
- [16 April 2022 - Correction](#): [aria-checked](#) is not to be used on elements that support the `checked` attribute.
- [03 April 2022 - Addition](#): Identify [Naming Prohibited](#) elements.
- [06 March 2022 - Addition](#): Allow `none` and `presentation` roles on [nav element](#).
- [03 March 2022 - Addition](#): Restrict role allowances for [div element](#) when it is a child of a [dl](#) element.
- [12 February 2022 - Addition & Correction](#): Allow `combobox` role on [button element](#). Allow `combobox` and `checkbox` roles on [input type=button element](#).
- [18 January 2022 - Addition](#): Added [Requirements for deprecated ARIA role, state and property attributes](#).
- [06 January 2022 - Addition](#): Change allowances for `doc-biblioentry` and `doc-endnote` roles on the [li element](#). These roles are deprecated in [Digital Publishing WAI-ARIA Module 1.1](#).
- [13 December 2021 - Correction](#): Allow `radio` role on [img alt="some text" element](#).
- [07 December 2021 - Correction](#): Allow only `none` and `presentation` roles for [wbr element](#). Allow only `aria-hidden` global attribute for [br](#) and [wbr](#) elements.
- [02 December 2021 - Addition](#): Allow `group` role on [section element](#).

- [16 November 2021 - Addition](#): Allow `link` and `button` roles on [area without href element](#).
- [26 October 2021 - Addition](#): Allow `aria-hidden` attribute on the [picture element](#).

Reviewers of the document can identify candidate additions and/or corrections by their distinctive styling in the document:

Candidate corrections are marked in the document.

Candidate additions are marked in the document.

This document was published by the [Web Applications Working Group](#) as a Recommendation using the [Recommendation track](#). It includes [candidate amendments](#), introducing substantive changes and new features since the previous Recommendation.

W3C recommends the wide deployment of this specification as a standard for the Web.

A W3C Recommendation is a specification that, after extensive consensus-building, is endorsed by W3C and its Members, and has commitments from Working Group members to [royalty-free licensing](#) for implementations. Future updates to this Recommendation may incorporate [new features](#).

Candidate additions are marked in the document.

Candidate corrections are marked in the document.

This document was produced by a group operating under the [W3C Patent Policy](#). W3C maintains a [public list of any patent disclosures](#) made in connection with the deliverables of the group; that page also includes instructions for disclosing a patent. An individual who has actual knowledge of a patent that the individual believes contains [Essential Claim\(s\)](#) must disclose the information in accordance with [section 6 of the W3C Patent Policy](#).

This document is governed by the [03 November 2023 W3C Process Document](#).

Table of Contents

Abstract

Status of This Document

- 1. Author requirements for use of ARIA in HTML**
- 2. ARIA semantics that extend and diverge from HTML**
- 3. Author guidance to avoid incorrect use of ARIA**
 - 3.1 Avoid overriding interactive elements with non-interactive roles
 - 3.2 Avoid specifying redundant roles
 - 3.3 Be cautious of side effects
 - 3.4 Adhere to the rules of ARIA
 - 3.5 Adhere to the rules of HTML
- 4. Document conformance requirements for use of ARIA attributes in HTML**
 - 4.1 Requirements for use of ARIA attributes to name elements
 - 4.2 Requirements for use of ARIA attributes in place of equivalent HTML attributes
 - 4.3 Requirements for deprecated ARIA role, state and property and attributes
 - 4.4 Case requirements for ARIA role, state and property attributes
- 5. Allowed descendants of ARIA roles**
- 6. Conformance**
 - 6.1 Conformance checking requirements
- 7. Privacy and security considerations**
- A. References**
 - A.1 Normative references
 - A.2 Informative references

§ 1. Author requirements for use of ARIA in HTML

Authors *MAY* use the ARIA role and `aria-*` attributes to change the exposed meaning ([semantics](#)) of [HTML elements](#), in accordance with the requirements defined by [WAI-ARIA](#), except where ARIA features conflict with the [strong native semantics](#) or are equal to the [implicit ARIA semantics](#) of a given HTML element. The [implicit ARIA semantics](#) for the features of HTML are defined by the [HTML Accessibility API Mappings](#) specification.

Any constraints for the use of ARIA features in HTML defined by this specification are intended to prevent authors from making assistive technology products report nonsensical user interface (UI)

information that does not represent the actual UI of the document.

Authors *MUST NOT* use the ARIA `role` and `aria-*` attributes in a manner that conflicts with the semantics described in the [4. Document conformance requirements for use of ARIA attributes in HTML](#) and [4.2 Requirements for use of ARIA attributes in place of equivalent HTML attributes](#) tables. It is *NOT RECOMMENDED* for authors to set the ARIA `role` and `aria-*` attributes to values that match the [implicit ARIA semantics](#) defined in either table. Doing so is unnecessary and can potentially lead to unintended consequences.

§ 2. ARIA semantics that extend and diverge from HTML

This section is non-normative.

Through the use of ARIA, authors can specify semantics that go beyond the current capabilities of native HTML. This can be very useful, as it provides authors the opportunity to create widgets, or expose specific accessible states and properties to native HTML features which would not be possible by the use of HTML alone.

For instance, a `button` element has no native HTML feature to expose a "pressed" state. ARIA allows authors to extend the semantics of the element by specifying the `aria-pressed` attribute, allowing for an aural UI that will match the visual presentation of the control.

In the following example, a `button` element allows for a user to toggle the state of a setting within a web application. The `aria-pressed` attribute is used to augment the `button` element. When in the "pressed" state that information can be exposed to users of assistive technologies.

EXAMPLE 1: Communicate a button's pressed state with ARIA

```
<button aria-pressed=true>...</button>
```

There are also situations where certain `aria-*` attributes are allowed for use on elements with specific roles, while the equivalent native attribute is currently not valid in HTML itself.

For instance, HTML has no direct concept of a disabled hyperlink (`a href` element). Constructs such as `<a href="..." disabled> ... </a>` are not valid, and will not be conveyed to assistive technologies.

ARIA diverges from HTML in this regard and does allow for an `aria-disabled` attribute to be

specified on an element with an explicit `role=link`. If an author were to specify an `aria-disabled=true` on an HTML hyperlink, user agents would not functionally treat the hyperlink any differently (it would still be clickable/operable), however it would be exposed to assistive technologies as being in the disabled state.

Similarly, while native HTML `option` elements that are descendants of a `select` can only be set as being `selected`, elements with an explicit `option` role can not only allow the equivalent `aria-selected`, but also the `aria-checked` attribute, supporting widgets/constructs that go beyond the capabilities of a native `select` element.

Unfortunately, in these situations where ARIA and HTML have feature parity, but diverge in allowances, it can create for a misalignment in support, if not also user experiences. In situations where ARIA allows a feature not supported by HTML, it will often be in the author's and ultimately the user's best interest to instead implement as a fully custom ARIA widget.

In the following example, a hyperlink needs to be communicated as being in the disabled state. HTML does not allow for the use of the `disabled` attribute on a hyperlink, and using `aria-disabled=true` would communicate the hyperlink as being disabled to assistive technologies, but would not actually disable the element. The most effective way to both communicate and actually disable a hyperlink would be to remove the `href` from the `a` element, creating a placeholder. Then, ARIA can be applied to this placeholder link to communicate the element's intended role and state.

EXAMPLE 2: Communicate a disabled link with ARIA

```
<a role=link aria-disabled=true>...</a>
```

§ 3. Author guidance to avoid incorrect use of ARIA

This section is non-normative.

§ 3.1 Avoid overriding interactive elements with non-interactive roles

ARIA is useful for revising or correcting the role of an element when a different role is necessary to expose to users. However, it is rarely in the user or author's best interest to try and use ARIA to

override an interactive element, for instance a `button`, with a role generally exposed by a non-interactive element. For instance, a heading.

As an example, the following uses a `role=heading` on a `button` element. This is not allowed, because the `button` element has default functionality that conflicts with user expectations for the heading role.

EXAMPLE 3: Wrong role

```
<button role="heading">search</button>
```

An author would need to take additional steps to ensure the default functionality and presentation of the `button` was removed, and even doing so may still not be enough to fully suppress the element's implicit features depending on how the user chooses to engage with the web page. E.g., by turning on Windows high contrast themes, or viewing the web page in a browser's reader mode.

§ 3.2 Avoid specifying redundant roles

The following example illustrates a `button` element which has also been provided an explicit `role=button`. Specifying this role is unnecessary, as a "button" element is already exposed with an implicit `button` role. In practice this particular instance of redundancy will likely not have unforeseen side effects, other than unnecessarily making the markup more verbose, and incorrectly signaling to other authors that this practice is useful. Please review the section [3.3 Be cautious of side effects](#) for an example of where specifying unnecessary roles can be problematic.

EXAMPLE 4: Redundant role on button

```
<!-- Avoid doing this! -->  
<button role="button">...</button>
```

Similarly, the following uses a `role=group` on a `fieldset` element, and a `role=Main` on a `main` element. This is unnecessary, because the `fieldset` element is implicitly exposed as a `role=group`, as is the `main` element implicitly exposed as a `role=main`. Again, in practice this will likely not have any unforeseen side effects to users of assistive technology, as long as the declaration of the `role` value uses [ASCII lowercase](#). Please see [4.4 Case requirements for ARIA role, state and property](#)

[attributes](#) for more information.

EXAMPLE 5: Redundant role on fieldset and main

```
<!-- Avoid doing this! -->
<fieldset role="group">...</fieldset>
<!-- or this! -->
<main role="Main">...</main>
```

The following uses a `role=list` on an [ul](#) element. As the `ul` element has an implicit role of `list`, explicitly adding the role would generally be considered redundant. However, some user agents suppress a list's implicit ARIA semantics if the list markers are removed from the visual presentation of the list items. Generally the redundant declaration of an element's implicit role would not be recommended, but in specific situations such as this, and where the role is necessary to expose, authors can explicitly add the role.

EXAMPLE 6: Redundant role on list

```
<!-- Generally avoid doing this! -->
<ul role="list">...</ul>
```

§ 3.3 Be cautious of side effects

The following uses a `role=button` on a [summary](#) element. This is unnecessary and can result in cross-platform issues. For instance, preventing the element from correctly exposing its state, and forcing the role of `button`, when it might otherwise be exposed with a platform or browser specific role.

EXAMPLE 7: Unintended consequences

```
<details>
  <!-- Avoid doing this! -->
  <summary role="button">more information</summary>
  ...
</details>
```

§ 3.4 Adhere to the rules of ARIA

[Accessible Rich Internet Applications \(WAI-ARIA\) 1.2](#) defines a number of roles which are not meant to be used by authors. Many of these roles are categorized as [Abstract Roles](#) which are explicitly stated as not to be used by authors. The following example illustrates the invalid use of an abstract select role, where an author likely meant to use the combobox role instead.

EXAMPLE 8: Abstract roles are not for authors

```
<!-- Do not do this! -->
<div role="select" ...>...</div>
```

ARIA also defines a [generic role](#) which is meant to provide feature parity with a number of HTML elements that do not have more specific ARIA semantics of their own. For instance, HTML's [div](#) and [span](#) elements, among others. ARIA discourages authors from using the generic role as its intended purpose is for use by implementors of user agents.

In the following example, rather than using a generic role, authors are advised to use a div in place of the article element. If changing the HTML element is not possible, specifying a role of presentation or none would be acceptable alternatives to remove the implicit role of the article.

EXAMPLE 9: Do not specify elements as generic

```
<!-- Avoid doing this! -->
<article role="generic" ...>...</article>
```

Additionally, ARIA specifically mentions in [Conflicts with Host Language Semantics](#) that if authors use both native HTML features for exposing states and properties as well as their ARIA counterparts, then the host language features take priority over the explicit ARIA attributes that are also used.

For instance, in the following example an author is using HTML's `input type=checkbox` and has specified an `aria-checked=true`. However, user agents are meant to ignore the `aria-checked` attribute. Instead user agents would expose the state based on the native features of the form control.

EXAMPLE 10: The implicit checked state takes precedent over the explicit ARIA attribute

```
<!-- Do not do this! -->
<input type="checkbox" checked aria-checked="false">
```

§ 3.5 Adhere to the rules of HTML

While ARIA can be used to alter the way HTML features are exposed to users of assistive technologies, these modifications do not change the underlying parsing and allowed content models of HTML. For instance, a [div](#) allows an author to specify any role on it. However, this does not mean that the element can then be used in a way that deviates from the rules HTML has defined for the element.

For instance, in the following example an author has specified a role of `link` on a `div` element. While HTML allows for a hyperlink (exposed as a `role=link`) to be a descendant of a `p` element, the HTML parser does not allow a `div` to be a descendant of a `p` element.

EXAMPLE 11: Revised ARIA semantics with invalid HTML nesting

```
<!-- Do not do this! -->
<p>
  ... <div role=link tabindex=0>...</div> ...
</p>
```

The HTML parser will modify the above markup to be output as the following:

EXAMPLE 12: Unwanted rendered markup with valid alternative solution

```

<!-- The previous example's markup will render as follows -->
<p>...</p>
<div role=link tabindex=0>...</div>
...
<p></p>

<!-- Instead of a div, use a span. Spans are allowed descendants of p ele
<p>
  ... <span role=link tabindex=0>...</span> ...
</p>

```

While this specification indicates the allowed ARIA attributes that can be specified on each HTML element, this example illustrates that even if a role is allowed, the context in which it is used can still result in potential rendering and accessibility issues.

§ 4. Document conformance requirements for use of ARIA attributes in HTML

The following table provides normative per-element document conformance requirements for the use of ARIA markup in HTML documents. Additionally, it identifies the [implicit ARIA semantics](#) that apply to [HTML elements](#). The [implicit ARIA semantics](#) of these elements are defined in [HTML AAM](#).

Each language feature (element) in a cell in the first column implies the ARIA semantics (role, states, and properties) given in the cell in the second column of the same row. The third cell in each row defines the ARIA `role` values and `aria-*` attributes which authors *MAY* specify on the element. Where a cell in the third column includes the term **Any role** it indicates that any `role` value MAY be used on the element. However, it is *NOT RECOMMENDED* for authors to specify the implicit role of the element, the generic role, or a role [deprecated by ARIA](#) on these elements. If a cell in the third column includes the term **No role** it indicates that authors *MUST NOT* overwrite the implicit ARIA semantics, or native semantics of the HTML element.

WAI-ARIA identifies roles which have [prohibited states and properties](#). These roles do not allow certain WAI-ARIA attributes to be specified by authors. HTML elements which expose these implicit WAI-ARIA roles also prohibit authors from specifying these WAI-ARIA attributes.

Elements which are identified as *Naming prohibited* are those which authors *MUST NOT* specify an `aria-label` or `aria-labelledby` attribute, unless the element allows for its implicit role to be overwritten by an explicit WAI-ARIA role which allows naming from authors. For more information see [4.1 Requirements for use of ARIA attributes to name elements](#).

NOTE

While setting an ARIA `role` and/or `aria-*` attribute that matches the implicit ARIA semantics is *NOT RECOMMENDED*, in some situations explicitly setting these attributes can be helpful – for instance, for user agents that do not expose implicit ARIA semantics for certain elements.

NOTE

While it is conforming to use [Digital Publishing WAI-ARIA Module 1.0](#) `role` values as outlined in the following table, the use of these roles is not intended for implementation of websites. If using these role for purposes beyond the scope of the digital publishing industry, further manual testing will be necessary to ensure the intended experience is provided to users.

Rules of ARIA attribute usage by HTML element

HTML element	Implicit ARIA semantics (explicitly assigning these in markup is <i>NOT</i> <i>RECOMMENDED</i>)	ARIA role, state and property allowances
a with href	role= link	Roles: button, checkbox, menuitem, menuitemcheckbox, menuitemradio, option, radio, switch, tab or treeitem. (link is also allowed, but <i>NOT</i> <i>RECOMMENDED</i>.) Dpub Roles: doc-backlink, doc- biblioref, doc-glossref or doc-noteref Global aria-* attributes and any aria-* attributes applicable to the allowed roles. It is <i>NOT RECOMMENDED</i> to use aria-disabled="true" on an a element with an href attribute. NOTE If a link needs to be programmatically communicated as "disabled", remove the href attribute.
		Any role, though generic <i>SHOULD NOT</i> be used. Naming Prohibited

a without href	<code>role=generic</code>	Otherwise, global <code>aria-*</code> attributes and any <code>aria-*</code> attributes applicable to the allowed roles.
abbr	No corresponding role	Any role Naming Prohibited Otherwise, global <code>aria-*</code> attributes and any <code>aria-*</code> attributes applicable to the allowed roles.
address	<code>role=group</code>	Any role, though group is <i>NOT RECOMMENDED</i>. Otherwise, global <code>aria-*</code> attributes and any <code>aria-*</code> attributes applicable to the allowed roles.
area with href	<code>role=link</code>	No role other than link, which is <i>NOT RECOMMENDED</i>. Global <code>aria-*</code> attributes and any <code>aria-*</code> attributes applicable to the link role.
area without href	<code>role=generic</code>	Roles: button or link. (generic is also allowed, but <i>SHOULD NOT</i> be used.) Naming Prohibited

		Otherwise, global aria-* attributes and any aria-* attributes applicable to the allowed roles.
article	role= article	Roles: application, document, feed, main, none, presentation or region. (article is also allowed, but <i>NOT RECOMMENDED</i>.) Global aria-* attributes and any aria-* attributes applicable to the allowed roles.
aside	role= complementary	Roles: feed, none, note, presentation, region or search. (complementary is also allowed, but <i>NOT RECOMMENDED</i>.) DPub Roles: doc-dedication, doc-example, doc-footnote, doc-glossary, doc-pullquote or doc-tip Global aria-* attributes and any aria-* attributes applicable to the allowed roles.
audio	No corresponding role	Role: application Global aria-* attributes and any aria-* attributes applicable to the application role.

<u>autonomous custom element</u>	Role exposed from author defined <u>ElementInternals</u> Otherwise <code>role=<u>generic</u></code>	If role defined by <u>ElementInternals</u>, <u>no role</u> Otherwise, <u>any role</u>, though <u>generic</u> <i>SHOULD NOT</i> be used. <u>Naming Prohibited</u> if exposed as the <u>generic</u> role, or if exposed as another role which prohibits naming. Otherwise, <u>global aria-* attributes</u> and any <u>aria-*</u> attributes applicable to the allowed roles.
<u>b</u>	<code>role=<u>generic</u></code>	<u>Any role</u>, though <u>generic</u> <i>SHOULD NOT</i> be used. <u>Naming Prohibited</u> Otherwise, <u>global aria-* attributes</u> and any <u>aria-*</u> attributes applicable to the allowed roles.
<u>base</u>	<u>No corresponding role</u>	<u>No role</u> or <u>aria-*</u> attributes
<u>bdi</u>	<code>role=<u>generic</u></code>	<u>Any role</u>, though <u>generic</u> <i>SHOULD NOT</i> be used. <u>Naming Prohibited</u> Otherwise, <u>global aria-* attributes</u> and any <u>aria-*</u> attributes applicable to the allowed roles.

		roles.
<u>bdo</u>	role= <u>generic</u>	<u>Any role</u>, though <u>generic</u> <i>SHOULD NOT</i> be used. <u>Naming Prohibited</u> Otherwise, <u>global aria-*</u> <u>attributes</u> and any <u>aria-*</u> attributes applicable to the allowed roles.
<u>blockquote</u>	role= <u>blockquote</u>	<u>Any role</u>, though <u>blockquote</u> is <i>NOT RECOMMENDED</i>. <u>Global aria-*</u> <u>attributes</u> and any <u>aria-*</u> attributes applicable to the allowed roles.
<u>body</u>	role= <u>generic</u>	<u>No role</u> other than <u>generic</u>, which <i>SHOULD NOT</i> be used. <u>Naming Prohibited</u> Otherwise, <u>global aria-*</u> <u>attributes</u> <u>are</u> allowed for the <u>generic</u> role, with the exception that authors <i>MUST NOT</i> specify <u>aria-hidden=true</u> on the body element.
<u>br</u>	<u>No corresponding role</u>	Roles: <u>none</u> or <u>presentation</u> Authors <i>MAY</i> specify the <u>aria-hidden</u> attribute on the <u>br</u> element. Otherwise, no other

		allowed <code>aria-*</code> attributes.
<u>button</u>	role=<u>button</u> If the button is the first child of a select element, the element is inert.	Roles: <u>checkbox</u>, <u>combobox</u>, <u>gridcell</u>, <u>link</u>, <u>menuitem</u>, <u>menuitemcheckbox</u>, <u>menuitemradio</u>, <u>option</u>, <u>radio</u>, <u>separator</u>, <u>slider</u>, <u>switch</u>, <u>tab</u>, or <u>treeitem</u>. (<u>button</u> is also allowed, but <i>NOT RECOMMENDED</i>.) <u>Global <code>aria-*</code> attributes</u> and any <code>aria-*</code> attributes applicable to the allowed roles. ✧ ✧ ✧ If the button is the first child of a select element: No role or <code>aria-*</code> attributes
<u>canvas</u>	<u>No corresponding role</u>	<u>Any role</u> <u>Global <code>aria-*</code> attributes</u> and any <code>aria-*</code> attributes applicable to the allowed roles.
<u>caption</u>	role=<u>caption</u>	<u>No role</u> other than <u>caption</u>, which is <i>NOT RECOMMENDED</i>. <u>Naming Prohibited</u> Otherwise, <u>global <code>aria-*</code> attributes</u>.

cite	No corresponding role	Any role Naming Prohibited Otherwise, global aria-* attributes and any aria-* attributes applicable to the allowed roles.
code	role= code	Any role, though code is <i>NOT RECOMMENDED</i>. Naming Prohibited Otherwise, global aria-* attributes and any aria-* attributes applicable to the allowed roles.
col	No corresponding role	No role or aria-* attributes
colgroup	No corresponding role	No role or aria-* attributes
data	role= generic	Any role, though generic <i>SHOULD NOT</i> be used. Naming Prohibited Otherwise, global aria-* attributes and any aria-* attributes applicable to the allowed roles.

<u>datalist</u>	role= <u>listbox</u>	<u>No role</u> other than <u>listbox</u>, which is <i>NOT RECOMMENDED</i>. No aria-* attributes.
<u>dd</u>	<u>No corresponding role</u>	<u>No role</u> Global <u>aria-* attributes</u> and any aria-* attributes applicable to the definition role.
<u>del</u>	role= <u>deletion</u>	<u>Any role</u>, though <u>deletion</u> is <i>NOT RECOMMENDED</i>. <u>Naming Prohibited</u> Otherwise, <u>global aria-* attributes</u> and any aria-* attributes applicable to the allowed roles.
<u>details</u>	role= <u>group</u>	<u>No role</u> other than <u>group</u>, which is <i>NOT RECOMMENDED</i>. Global <u>aria-* attributes</u> and any aria-* attributes applicable to the group role.
<u>dfn</u>	role= <u>term</u>	<u>Any role</u>, though <u>term</u> is <i>NOT RECOMMENDED</i>. Global <u>aria-* attributes</u> and any aria-* attributes applicable to the allowed roles.

dialog	role= dialog	Role: alertdialog. (dialog is also allowed, but <i>NOT RECOMMENDED</i>.) Global aria-* attributes and any aria-* attributes applicable to the dialog role.
div	role= generic	If a direct child of a dl element, only presentation or none. Otherwise, any role, though generic <i>SHOULD NOT</i> be used. Naming Prohibited Otherwise, global aria-* attributes and any aria-* attributes applicable to the allowed roles.
dl	No corresponding role	Roles: group, list, none or presentation Global aria-* attributes and any aria-* attributes applicable to the allowed roles.
dt	No corresponding role	Role: listitem Global aria-* attributes and any aria-* attributes applicable to the allowed roles.
		Any role, though emphasis is

em	role= emphasis	NOT RECOMMENDED. Naming Prohibited Otherwise, global aria-* attributes and any aria-* attributes applicable to the allowed roles.
embed	No corresponding role	Roles: application, document, img, image, none or presentation. Global aria-* attributes and any aria-* attributes applicable to the allowed roles.
fieldset	role= group	Roles: none, presentation or radiogroup. (group is also allowed, but <i>NOT RECOMMENDED</i>.) Global aria-* attributes and any aria-* attributes applicable to the allowed roles.
figcaption	No corresponding role	Roles: group, none or presentation Naming Prohibited Otherwise, global aria-* attributes and any aria-* attributes applicable to the allowed roles.

figure	<code>role=figure</code> If the figure has a valid <code>figcaption</code> descendant: No role other than figure, which is <i>NOT RECOMMENDED</i>. Dpub Role: doc-example. Otherwise, if the figure has no <code>figcaption</code> descendant: Any role, though figure is <i>NOT RECOMMENDED</i>. Global aria-* attributes and any <code>aria-*</code> attributes applicable to the allowed roles.
footer	If not a descendant of an <code>article</code>, <code>aside</code>, <code>main</code>, <code>nav</code> or <code>section</code> element, or an element with <code>role=article</code>, <code>complementary</code>, <code>main</code>, <code>navigation</code> or <code>region</code> then <code>role=contentinfo</code> Otherwise, <code>role=generic</code> Roles: group, presentation or none. (If not a descendant of an <code>article</code>, <code>aside</code>, <code>main</code>, <code>nav</code> or <code>section</code> element, or an element with <code>role=article</code>, <code>complementary</code>, <code>main</code>, <code>navigation</code> or <code>region</code>, then <code>role=contentinfo</code> is also allowed, but <i>NOT RECOMMENDED</i>. Otherwise, <code>role=generic</code> is also allowed, but <i>SHOULD NOT</i> be used.) Dpub Role: doc-footnote Naming Prohibited if exposed as <code>generic</code>. Otherwise, global aria-* attributes and any <code>aria-*</code> attributes applicable to the allowed roles.

form	role=form	Roles: none, presentation or search. (form is also allowed, but <i>NOT RECOMMENDED</i>.) Global aria-* attributes and any aria-* attributes applicable to the allowed roles. NOTE A form is not exposed as a landmark region unless it has been provided an accessible name.
form-associated custom element	Role exposed from author defined ElementInternals Otherwise role=generic	If role defined by ElementInternals, no role Otherwise, form-related roles: button, checkbox, combobox, listbox, progressbar, group, radio, radiogroup, searchbox, slider, spinbutton, switch or textbox. (generic is also allowed, but <i>SHOULD NOT</i> be used.) Naming Prohibited if exposed as the generic role. Otherwise, global aria-* attributes and any aria-* attributes applicable to the allowed roles.
		Roles: none, presentation or

h1 to h6	role= heading , aria-level = the number in the element's tag name	tab . (heading is also allowed, but <i>NOT RECOMMENDED</i>). Dpub Role: doc-subtitle Global aria-* attributes and any aria-* attributes applicable to the allowed roles.
head	No corresponding role	No role or aria-* attributes
header	If not a descendant of an article, aside, main, nav or section element, or an element with role=article, complementary, main, navigation or region then role=banner Otherwise, role=generic	Roles: group, none or presentation. (If not a descendant of an article, aside, main, nav or section element, or an element with role=article, complementary, main, navigation or region, then role=banner is also allowed, but <i>NOT RECOMMENDED</i>. Otherwise, role=generic is also allowed, but <i>SHOULD NOT</i> be used.) Naming Prohibited if exposed as generic. Otherwise, global aria-* attributes and any aria-* attributes applicable to the allowed roles.
hgroup	role= group	Any role, though group is <i>NOT RECOMMENDED</i>. Otherwise, global aria-* attributes and any aria-*

		attributes applicable to the allowed roles.
hr	role= separator	Roles: none or presentation. (separator is also allowed, but <i>NOT RECOMMENDED</i>.) Dpub Role: doc-pagebreak Global aria-* attributes and any aria-* attributes applicable to the separator role.
html	role= generic	No role other than document or generic, which are <i>NOT RECOMMENDED</i>. No aria-* attributes.
i	role= generic	Any role, though generic <i>SHOULD NOT</i> be used. Naming Prohibited Otherwise, global aria-* attributes and any aria-* attributes applicable to the allowed roles.
iframe	No corresponding role	Roles: application, document, img, image, none or presentation. Global aria-* attributes and any aria-* attributes applicable to the allowed roles.

img

If the `img` has non-empty `alt` (`alt="some text"`) or an accessible name is provided by another [img naming method](#), or the `img` has no `alt` and has not been provided a name:
`role=`[img or image](#)

Roles: [button](#), [checkbox](#), [link](#), [math](#), [menuitem](#), [menuitemcheckbox](#), [menuitemradio](#), [meter](#), [option](#), [progressbar](#), [radio](#), [scrollbar](#), [separator](#), [slider](#), [switch](#), [tab](#) or [treeitem](#). ([img or image](#) is also allowed, but *NOT RECOMMENDED*.)

Dpub Role: [doc-cover](#)

[Global aria-* attributes](#) and any `aria-*` attributes applicable to the allowed roles.

[img](#) with no accessible name

If the `img` has an empty `alt` (`alt=""`) and lacks any other [img naming methods](#):
`role=`[none](#),
`role=`[presentation](#)

If the `img` [lacks an alt attribute](#) and lacks any other [img naming methods](#):
`role=`[img or image](#)

If the `img` has no `alt` attribute or accessible name: **[No role](#)** other than the `role=`[none](#) or [presentation](#) roles.
(`role=`[img or image](#) is also allowed, but *NOT RECOMMENDED*.)

If the `img` has an empty `alt=""` attribute and no `aria-label` or `aria-labelledby` attributes to provide it an accessible name: **[No role](#)** other than the `role=`[none](#) or [presentation](#) roles, which are *NOT RECOMMENDED*.

No `aria-*` attributes except [aria-hidden="true"](#).

Otherwise, if the `img` has an

		author defined accessible name, see img with an accessible name.
<u>input</u> <u>type=button</u>	role= <u>button</u>	Roles: <u>checkbox</u>, <u>combobox</u>, <u>gridcell</u>, <u>link</u>, <u>menuitem</u>, <u>menuitemcheckbox</u>, <u>menuitemradio</u>, <u>option</u>, <u>radio</u>, <u>separator</u>, <u>slider</u>, <u>switch</u>, <u>tab</u>, or <u>treeitem</u>. (<u>button</u> is also allowed, but <i>NOT RECOMMENDED</i>.) Global aria-* attributes and any aria-* attributes applicable to the allowed roles.
<u>input</u> <u>type=checkbox</u>	role= <u>checkbox</u>	Roles: <u>menuitemcheckbox</u>, <u>option</u> or <u>switch</u>; <u>button</u> if used with aria-pressed. (<u>checkbox</u> is also allowed, but <i>NOT RECOMMENDED</i>.) Authors <i>MUST NOT</i> use the aria-checked attribute on <code>input type=checkbox</code> elements. Otherwise, any global aria-* attributes and any aria-* attributes applicable to the allowed roles. NOTE The HTML checked attribute can be used instead

		of the aria-checked attribute for menuitemcheckbox, option or switch roles when used on type=checkbox.
<u>input type=color</u>	<u>No corresponding role</u>	<u>No role</u> <u>Global aria-* attributes</u> and aria-disabled attribute.
<u>input type=date</u>	<u>No corresponding role</u>	<u>No role</u> <u>Global aria-* attributes</u> and any aria-* attributes applicable to the textbox role.
<u>input type=datetime-local</u>	<u>No corresponding role</u>	<u>No role</u> <u>Global aria-* attributes</u> and any aria-* attributes applicable to the textbox role.
<u>input type=email</u> with no <u>list</u> attribute	role= <u>textbox</u>	<u>No role</u> other than <u>textbox</u> , which is <i>NOT RECOMMENDED</i> . <u>Global aria-* attributes</u> and any aria-* attributes applicable to the textbox role.
<u>input type=file</u>	<u>No corresponding role</u>	<u>No role</u> <u>Global aria-* attributes</u> , aria-disabled, aria-invalid and

	aria-required attributes.	
<u>input type=hidden</u>	<u>No corresponding role</u>	<u>No role</u> or aria-* attributes
<u>input type=image</u>	role= <u>button</u>	The following roles are allowed, but are <i>NOT RECOMMENDED</i>: <u>button</u>, <u>checkbox</u>, <u>gridcell</u>, <u>link</u>, <u>menuitem</u>, <u>menuitemcheckbox</u>, <u>menuitemradio</u>, <u>option</u>, <u>radio</u>, <u>separator</u>, <u>slider</u>, <u>switch</u>, <u>tab</u> or <u>treeitem</u>. <u>Global aria-* attributes</u> and any aria-* attributes applicable to the allowed roles. If possible, authors <i>SHOULD</i> consider using a different HTML element which allows the specified role, such as the button element.
<u>input type=month</u>	<u>No corresponding role</u>	<u>No role</u> <u>Global aria-* attributes</u> and any aria-* attributes applicable to the textbox role.
<u>input type=number</u>	role= <u>spinbutton</u>	<u>No role</u> other than <u>spinbutton</u> , which is <i>NOT RECOMMENDED</i> . <u>Global aria-* attributes</u> and any aria-* attributes applicable to the spinbutton role.

<u>input type=password</u>	<u>No corresponding role</u>	<u>No role</u> <u>Global aria-* attributes</u> and any <code>aria-*</code> attributes applicable to the <code>textbox</code> role.
<u>input type=radio</u>	<code>role=</code> <u>radio</u>	Role: <u>menuitemradio</u>. (<u>radio</u> is also allowed, but <i>NOT RECOMMENDED</i>.) Authors <i>MUST NOT</i> use the <u>aria-checked</u> attribute on <u>input type=radio</u> elements. Otherwise, any <u>global aria-* attributes</u> and any <code>aria-*</code> attributes applicable to the allowed roles. NOTE The HTML <u>checked</u> attribute can be used instead of the <code>aria-checked</code> attribute for the <code>menuitemradio</code> role when used on <code>type=radio</code>.
<u>input type=range</u>	<code>role=</code> <u>slider</u>	<u>No role</u> other than <u>slider</u>, which is <i>NOT RECOMMENDED</i>. Authors <i>SHOULD NOT</i> use the <u>aria-valuemax</u> or <u>aria-valuemin</u> attributes on <code>input type=range</code>.

		Otherwise, any global aria-* attributes and any other aria-* attributes applicable to the <code>slider</code> role.
input type=reset	role= button	The following roles are allowed, but are <i>NOT RECOMMENDED</i>: button, checkbox, combobox, gridcell, link, menuitem, menuitemcheckbox, menuitemradio, option, radio, separator, slider, switch, tab or treeitem. Global aria-* attributes and any aria-* attributes applicable to the allowed roles. If possible, authors <i>SHOULD</i> consider using a different HTML element which allows the specified role, such as the <code>button</code> element.
input type=search , with no list attribute	role= searchbox	No role other than searchbox, which is <i>NOT RECOMMENDED</i>. Global aria-* attributes and any aria-* attributes applicable to the <code>searchbox</code> role.
		The following roles are allowed, but are <i>NOT RECOMMENDED</i>: button,

<u>input type=submit</u>	role= button	checkbox, combobox, gridcell, link, menuitem, menuitemcheckbox, menuitemradio, option, radio, separator, slider, switch, tab or treeitem. Global aria-* attributes and any aria-* attributes applicable to the allowed roles. If possible, authors <i>SHOULD</i> consider using a different HTML element which allows the specified role, such as the button element.
<u>input type=tel</u> , with no list attribute	role= textbox	No role other than textbox, which is <i>NOT RECOMMENDED</i>. Global aria-* attributes and any aria-* attributes applicable to the textbox role.
<u>input type=text</u> or with a missing or invalid type , with no list attribute	role= textbox	Roles: combobox, searchbox or spinbutton. (textbox is also allowed, but <i>NOT RECOMMENDED</i>.) Global aria-* attributes and any aria-* attributes applicable to the allowed roles.
<u>input type=text</u> ,		No role other than combobox, which is <i>NOT RECOMMENDED</i>. Authors <i>SHOULD NOT</i> use the

search , tel , url , email , or with a missing or invalid type, with a list attribute	role= combobox	aria-haspopup attribute on the indicated inputs with a list attribute. Otherwise, any global aria-* attributes and any other aria-* attributes applicable to the combobox role.
input type=time	No corresponding role	No role Global aria-* attributes and any aria-* attributes applicable to the textbox role.
input type=url with no list attribute	role= textbox	No role other than textbox, which is <i>NOT RECOMMENDED</i>. Global aria-* attributes and any aria-* attributes applicable to the textbox role.
input type=week	No corresponding role	No role Global aria-* attributes and any aria-* attributes applicable to the textbox role.
ins	role= insertion	Naming Prohibited Otherwise, global aria-* attributes and any aria-* attributes applicable to the allowed roles

		roles.
<u>kbd</u>	<u>No corresponding role</u>	<u>Any role</u> <u>Naming Prohibited</u> Otherwise, <u>global aria-*</u> <u>attributes</u> and any <u>aria-*</u> attributes applicable to the allowed roles.
<u>label</u>	<u>No corresponding role</u>	If a <code>label</code> element is implicitly or explicitly associated with a <u>labelable element</u> then <u>no role</u> Otherwise, if the <code>label</code> is not associated with an element then <u>any</u> <u>role</u>, though <u>generic</u> <i>SHOULD</i> <i>NOT</i> be used. <u>Naming Prohibited</u> if exposed as the <code>generic</code> role, or if exposed as another role which prohibits naming. Otherwise, <u>global aria-*</u> <u>attributes</u>.
<u>legend</u>	<u>No corresponding role</u>	<u>No role</u> <u>Naming Prohibited</u> Otherwise, <u>global aria-*</u> <u>attributes</u>.
		<u>No role</u> other than <u>listitem</u>,

li	If the <code>li</code> is a child of a list element (ul, ol, menu) <code>role=listitem</code>. Otherwise, if the <code>li</code> is not a child of a list element it is exposed as a <code>role=generic</code>.	which is <i>NOT RECOMMENDED</i>, if the parent list element has an implicit or explicit <code>list</code> role. Otherwise, any role if the parent list element does not expose an implicit or explicit <code>list</code> role. NOTE See ul, ol, or menu for allowed roles for list elements. Global <code>aria-*</code> attributes and any <code>aria-*</code> attributes applicable to the allowed roles. Authors <i>SHOULD NOT</i> use the following deprecated DPub Roles: doc-biblioentry, doc-endnote.
link	No corresponding role	No role or <code>aria-*</code> attributes
main	<code>role=main</code>	No role other than main, which is <i>NOT RECOMMENDED</i>. Global <code>aria-*</code> attributes and any <code>aria-*</code> attributes applicable to the <code>main</code> role.
map	No corresponding role	No role or <code>aria-*</code> attributes

mark	No corresponding role	Any role Naming Prohibited Otherwise, global aria-* attributes and any aria-* attributes applicable to the allowed roles.
math	role= math	No role other than math, which is <i>NOT RECOMMENDED</i>. Global aria-* attributes and any aria-* attributes applicable to the math role.
menu	role= list	Roles: group, listbox, menu, menubar, none, presentation, radiogroup, tablist, toolbar or tree. (list is also allowed, but <i>NOT RECOMMENDED</i>.) Global aria-* attributes and any aria-* attributes applicable to the allowed roles. Authors <i>SHOULD NOT</i> use deprecated directory role.
meta	No corresponding role	No role or aria-* attributes No role other than meter, which is <i>NOT RECOMMENDED</i>.

meter	role= meter	Authors <i>SHOULD NOT</i> use the aria-valuemax or aria-valuemin attributes on meter elements. Otherwise, any global aria-* attributes and any other aria-* attributes applicable to the meter role.
nav	role= navigation	Roles: menu, menubar, none, presentation or tablist. (navigation is also allowed, but <i>NOT RECOMMENDED</i>.) Dpub Roles: doc-index, doc-pagelist or doc-toc Global aria-* attributes and any aria-* attributes applicable to the allowed roles.
noscript	No corresponding role	No role or aria-* attributes
object	No corresponding role	Roles: application, document, img or image. Global aria-* attributes and any aria-* attributes applicable to the allowed roles.
		Roles: group, listbox, menu, menubar, none, presentation, radiogroup, tablist, toolbar or tree. (list is also allowed, but <i>NOT RECOMMENDED</i>.)

ol	role= list	Global <code>aria-*</code> attributes and any <code>aria-*</code> attributes applicable to the allowed roles. Authors <i>SHOULD NOT</i> use deprecated <code>directory</code> role.
optgroup	role= group	No role other than group, which is <i>NOT RECOMMENDED</i>. Global <code>aria-*</code> attributes and any <code>aria-*</code> attributes applicable to the group role.
option element that is in a list of options or that represents a suggestion in a datalist	role= option	No role other than option, which is <i>NOT RECOMMENDED</i>. Authors <i>SHOULD NOT</i> use the <code>aria-selected</code> attribute on the option element. Global <code>aria-*</code> attributes and any other <code>aria-*</code> attributes applicable to the option role.
output	role= status	Any role, though status is <i>NOT RECOMMENDED</i>. Global <code>aria-*</code> attributes and any <code>aria-*</code> attributes applicable to the allowed roles.
		Any role, though paragraph is <i>NOT RECOMMENDED</i>. Naming Prohibited

p	role= paragraph	<u>Naming Prohibited</u> Otherwise, global aria-* attributes and any aria-* attributes applicable to the allowed roles.
param	<u>No corresponding role</u>	<u>No role</u> or aria-* attributes
picture	<u>No corresponding role</u>	<u>No role</u> Authors <i>MAY</i> specify the aria-hidden attribute on the picture element. Otherwise, no other allowed aria-* attributes.
pre	role= generic	<u>Any role</u> , though generic <i>SHOULD NOT</i> be used. <u>Naming Prohibited</u> Otherwise, global aria-* attributes and any aria-* attributes applicable to the allowed roles.
progress	role= progressbar	<u>No role</u> other than progressbar , which is <i>NOT RECOMMENDED</i> . Authors <i>SHOULD NOT</i> use the aria-valuemax attribute on progress elements. Otherwise, any global aria-* attributes and any other aria-*

		attributes applicable to the progressbar role.
<u>q</u>	role= <u>generic</u>	<u>Any role</u>, though <u>generic</u> <i>SHOULD NOT</i> be used. <u>Naming Prohibited</u> Otherwise, <u>global aria-*</u> <u>attributes</u> and any <u>aria-*</u> attributes applicable to the allowed roles.
<u>rp</u>	<u>No corresponding role</u>	<u>Any role</u> <u>Naming Prohibited</u> <u>Global aria-*</u> <u>attributes</u> and any <u>aria-*</u> attributes applicable to the allowed roles.
<u>rt</u>	<u>No corresponding role</u>	<u>Any role</u> <u>Naming Prohibited</u> Otherwise, <u>global aria-*</u> <u>attributes</u> and any <u>aria-*</u> attributes applicable to the allowed roles.
<u>ruby</u>	<u>No corresponding role</u>	<u>Any role</u> <u>Global aria-*</u> <u>attributes</u> and any <u>aria-*</u> attributes applicable to the allowed roles.

<u>s</u>	role= <u>deletion</u>	<u>Any role</u>, though <u>deletion</u> is <i>NOT RECOMMENDED</i>. <u>Naming Prohibited</u> Otherwise, <u>global aria-*</u> <u>attributes</u> and any <u>aria-*</u> attributes applicable to the allowed roles.
<u>samp</u>	role= <u>generic</u>	<u>Any role</u>, though <u>generic</u> <i>SHOULD NOT</i> be used. <u>Naming Prohibited</u> Otherwise, <u>global aria-*</u> <u>attributes</u> and any <u>aria-*</u> attributes applicable to the allowed roles.
<u>script</u>	<u>No corresponding role</u>	<u>No role</u> or <u>aria-*</u> attributes
<u>search</u>	role= <u>search</u>	Roles: <u>form</u>, <u>group</u>, <u>none</u>, <u>presentation</u> or <u>region</u>. (<u>search</u> is also allowed, but <i>NOT RECOMMENDED</i>.) <u>Global aria-*</u> <u>attributes</u> and any <u>aria-*</u> attributes applicable to the allowed roles.
		Roles: <u>alert</u> . <u>alertdialog</u> .

	application, banner, complementary, contentinfo, dialog, document, feed, group, log, main, marquee, navigation, none, note, presentation, search, status or tabpanel. (role=region is also allowed, but <i>NOT RECOMMENDED</i>. role=generic <i>SHOULD NOT</i> be used.) DPub Roles: doc-abstract, doc-acknowledgments, doc-afterword, doc-appendix, doc-bibliography, doc-chapter, doc-colophon, doc-conclusion, doc-credit, doc-credits, doc-dedication, doc-endnotes, doc-epigraph, doc-epilogue, doc-errata, doc-example, doc-foreword, doc-glossary, doc-index, doc-introduction, doc-notice, doc-pagelist, doc-part, doc-preface, doc-prologue, doc-pullquote, doc-qna, doc-toc Global aria-* attributes and any aria-* attributes applicable to the allowed roles.
select (with NO multiple attribute and NO size	Role: menu. (combobox is also allowed, but <i>NOT RECOMMENDED</i>.) Authors <i>SHOULD NOT</i> use the aria-multiselectable attribute

attribute having value greater than 1)	role= combobox	on a select element. Otherwise, any global aria-* attributes and any other aria-* attributes applicable to the combobox or menu role.
select (with a multiple attribute or a size attribute having value greater than 1)	role= listbox	No role other than listbox, which is <i>NOT RECOMMENDED</i>. Authors <i>SHOULD NOT</i> use the aria-multiselectable attribute on a select element. Otherwise, any global aria-* attributes and any other aria-* attributes applicable to the listbox role.
selectedcontent	role= generic	If used as a valid descendant of a select element: no role or aria-* attributes Otherwise, any role if the element is used outside of its intended context as a child of the button part of a customizable select element, though generic is <i>NOT RECOMMENDED</i>. Naming Prohibited Otherwise, global aria-* attributes and any aria-* attributes applicable to the allowed roles.
slot	No corresponding role	No role or aria-* attributes

<u>small</u>	role= <u>generic</u>	<u>Any role</u>, though <u>generic</u> <i>SHOULD NOT</i> be used. <u>Naming Prohibited</u> Otherwise, <u>global aria-* attributes</u> and any aria-* attributes applicable to the allowed roles.
<u>source</u>	<u>No corresponding role</u>	<u>No role</u> or aria-* attributes
<u>span</u>	role= <u>generic</u>	<u>Any role</u>, though <u>generic</u> <i>SHOULD NOT</i> be used. <u>Naming Prohibited</u> Otherwise, <u>global aria-* attributes</u> and any aria-* attributes applicable to the allowed roles.
<u>strong</u>	role= <u>strong</u>	<u>Any role</u>, though <u>strong</u> is <i>NOT RECOMMENDED</i>. <u>Naming Prohibited</u> Otherwise, <u>global aria-* attributes</u> and any aria-* attributes applicable to the allowed roles.
<u>style</u>	<u>No corresponding role</u>	<u>No role</u> or aria-* attributes

<u>sub</u>	role= <u>subscript</u>	<u>Any role</u>, though <u>subscript</u> is <i>NOT RECOMMENDED</i>. <u>Naming Prohibited</u> Otherwise, <u>global aria-* attributes</u> and any aria-* attributes applicable to the allowed roles.
<u>summary</u>	<u>No corresponding role</u> NOTE Many, but not all, user agents expose the summary element with an implicit ARIA role=<u>button</u>.	<u>No role</u> if the summary element is a <u>summary for its parent details</u>. <u>Global aria-* attributes</u>, aria-disabled, and aria-haspopup attributes. Otherwise, authors <i>MAY</i> specify <u>Any role</u>, and any <u>global aria-* attributes</u> and any aria-* attributes applicable to the allowed roles.
<u>sup</u>	role= <u>superscript</u>	<u>Any role</u>, though <u>superscript</u> is <i>NOT RECOMMENDED</i>. <u>Naming Prohibited</u> Otherwise, <u>global aria-* attributes</u> and any aria-* attributes applicable to the allowed roles.

SVG	role=graphics-document as defined by SVG AAM	Any role, though graphics-document is <i>NOT RECOMMENDED</i>. Global aria-* attributes and any aria-* attributes applicable to the allowed roles.
table	role=table	Any role, though table is <i>NOT RECOMMENDED</i>. Global aria-* attributes and any aria-* attributes applicable to the allowed roles.
tbody	role=rowgroup	Any role, though rowgroup is <i>NOT RECOMMENDED</i>. Global aria-* attributes and any aria-* attributes applicable to the allowed roles.
td	role=cell if the ancestor table element is exposed as a role=table role=gridcell if the ancestor table element is exposed as a role=grid or treegrid No corresponding role if the ancestor table element is not exposed as a	If the ancestor table element has role=table, grid, or treegrid, no role other than the following: • If the ancestor table element is exposed as a role=table, then cell is allowed, but <i>NOT RECOMMENDED</i>. • If the ancestor table element is exposed as a role=grid or treegrid, then gridcell is allowed, but <i>NOT RECOMMENDED</i>. Otherwise, if the ancestor table

	not exposed as a role=table, grid or treegrid	element is not exposed as a role=table, grid or treegrid, any role . Global aria-* attributes and any aria-* attributes applicable to the allowed roles.
template	No corresponding role	No role or aria-* attributes
textarea	role= textbox	No role other than textbox , which is <i>NOT RECOMMENDED</i> . Global aria-* attributes and any aria-* attributes applicable to the textbox role.
tfoot	role= rowgroup	Any role , though rowgroup is <i>NOT RECOMMENDED</i> . Global aria-* attributes and any aria-* attributes applicable to the allowed roles.
th	role= columnheader , rowheader or cell if the ancestor table element is exposed as a role=table role= columnheader , rowheader or gridcell if the ancestor table element is exposed as a role=grid	If the ancestor table element has role=table, grid, or treegrid, no role other than the following: • If the ancestor table element is exposed as a role=table, then columnheader, rowheader and cell are allowed, but <i>NOT RECOMMENDED</i>. • If the ancestor table element is exposed as a role=grid or treegrid, then columnheader,

	is exposed as a <code>role=grid</code> or <code>treegrid</code> <u>No corresponding role</u> if the ancestor <code>table</code> element is not exposed as a <code>role=table</code>, <code>grid</code> or <code>treegrid</code>	<u>rowheader</u> or <u>gridcell</u> are allowed, but <i>NOT RECOMMENDED</i>. Otherwise, if the ancestor <code>table</code> element is not exposed as a <code>role=table</code>, <code>grid</code> or <code>treegrid</code>, <u>any role</u>. <u>Global <code>aria-*</code> attributes</u> and any <code>aria-*</code> attributes applicable to the allowed roles.
<u>thead</u>	<code>role=</code> <u>rowgroup</u>	<u>Any role</u>, though <u>rowgroup</u> is <i>NOT RECOMMENDED</i>. <u>Global <code>aria-*</code> attributes</u> and any <code>aria-*</code> attributes applicable to the allowed roles.
<u>time</u>	<code>role=</code> <u>time</u>	<u>Naming Prohibited</u> Otherwise, <u>global <code>aria-*</code> attributes</u> and any <code>aria-*</code> attributes applicable to the allowed roles.
<u>title</u>	<u>No corresponding role</u>	<u>No role</u> or <code>aria-*</code> attributes
		If the ancestor <code>table</code> element has <code>role=table</code>, <code>grid</code>, or <code>treegrid</code>, <u>no role</u> other than <u>row</u>, which is <i>NOT RECOMMENDED</i>; otherwise

tr	role= row	any role , though row is <i>NOT RECOMMENDED</i> . Global aria-* attributes and any aria-* attributes applicable to the allowed roles.
track	No corresponding role	No role or aria-* attributes
u	role= generic	Any role , though generic <i>SHOULD NOT</i> be used. Naming Prohibited Otherwise, global aria-* attributes and any aria-* attributes applicable to the allowed roles.
ul	role= list	Roles: group , listbox , menu , menubar , none , presentation , radiogroup , tablist , toolbar or tree . (list is also allowed, but <i>NOT RECOMMENDED</i> .) Global aria-* attributes and any aria-* attributes applicable to the allowed roles. Authors <i>SHOULD NOT</i> use the deprecated directory role.
		Any role Naming Prohibited

var	No corresponding role	Otherwise, global aria-* attributes and any aria-* attributes applicable to the allowed roles.
video	No corresponding role	Role: application Global aria-* attributes and any aria-* attributes applicable to the application role.
wbr	No corresponding role	Roles: none or presentation Authors <i>MAY</i> specify the aria-hidden attribute on the wbr element. Otherwise, no other allowed aria-* attributes.

The elements marked with *No corresponding role*, in the second column of the table do not have any [implicit ARIA semantics](#), but they do have meaning and this meaning may be represented in roles, states and properties not provided by ARIA, and exposed to users of assistive technology via accessibility APIs. It is therefore recommended that authors add a `role` attribute to a semantically neutral element such as a [div](#) or [span](#), rather than overriding the semantics of the listed elements.

NOTE

Authors are encouraged to make use of the following documents for guidance on using ARIA in HTML beyond that which is provided here:

- [Using ARIA](#) - A practical guide for authors on how to add accessibility information to HTML elements using the Accessible Rich Internet Applications specification.
- [WAI-ARIA Authoring Practices 1.2](#) - An author's guide to understanding and implementing Accessible Rich Internet Applications.

EXAMPLE 13

These features can be used to make accessibility tools render content to their users in more useful ways. For example, ASCII art, which is really an image, appears to be text, and in the absence of appropriate roles and properties would end up being rendered by screen readers as a nonsensical string of punctuation characters. Using the features described in this section, one can instead make the ATs skip the ASCII art and just read the caption:

```
<figure>
  <pre role="img" aria-label="ASCII Fish">
0      .` `/
  '    / (
0    .-'' `''-._ .')
    _/ (o)      ' . ' /
    )          )))  >< <
    ` \ | _ \    _.' ' \
      '-._ _.-'  '.)
    jgs    ` \ _ \
  </pre>
  <figcaption id="fish-caption">
    Joan G. Stark, "<cite>fish</cite>".
    October 1997. ASCII on electrons. 28x8.
  </figcaption>
</figure>
```

4.1 Requirements for use of ARIA attributes to name elements

Authors *MAY* use `aria-label` and `aria-labelledby` attributes to specify [accessible names](#) for elements which have an implicit or explicit ARIA role which allows naming from authors.

[Accessible Rich Internet Applications \(WAI-ARIA\) 1.2](#) defines [roles which allow naming from authors](#) as well as [roles where author naming is prohibited](#).

Authors *MUST NOT* specify `aria-label` or `aria-labelledby` on elements with implicit WAI-ARIA roles which cannot be named. HTML elements whose implicit WAI-ARIA roles prohibit naming from authors are identified in [4. Document conformance requirements for use of ARIA attributes in HTML](#).

The following markup example demonstrates a selection of HTML elements with implicit ARIA roles that prohibit naming from authors.

EXAMPLE 14: Elements with implicit ARIA roles which prohibit naming from authors

```
<!-- DO NOT do the following! -->
<p aria-label="...">...</p>

<span aria-label="...">...<span>

<code aria-label="...">...<code>

<div aria-labelledby="...">...</div>
```

The following markup example demonstrates elements which have explicit WAI-ARIA roles which allow naming from authors. Due to the explicit roles specified on these elements, `aria-label` and `aria-labelledby` attributes are allowed.

EXAMPLE 15: Elements with explicit ARIA roles which allow naming from authors

```
<p role="link" tabindex="0" aria-label="...">...</p>
```

```
<span role="button" tabindex="0" aria-label="...">...</span>
```

```
<div role="article" aria-labelledby="...">...</div>
```

§ 4.2 Requirements for use of ARIA attributes in place of equivalent HTML attributes

Unless otherwise stated, authors *MAY* use `aria-*` attributes in place of their HTML equivalents on HTML elements where the `aria-*` semantics would be expected. For example, authors *MAY* specify `aria-disabled=true` on a [button](#) element, while also implementing the necessary scripting to functionally disable the `button`, rather than the use `disabled` attribute.

As stated in [WAI-ARIA's Conflicts with Host Language Semantics](#), when HTML elements use *both* `aria-*` attributes and their host language (HTML) equivalents, user agents *MUST* ignore the WAI-ARIA attributes – the native HTML attributes with the same [implicit ARIA semantics](#) take precedence. For this reason, authors *SHOULD NOT* specify both the native HTML attribute and the equivalent `aria-*` attribute on an element. Please review each attribute for any further author specific requirements.

The following table represents HTML elements and their attributes which have `aria-*` attribute parity.

Each language feature (element and attribute) in a cell in the first column implies the ARIA semantics (states, and properties) given in the cell in the second column of the same row. The third cell in each row defines how authors can use the native HTML feature, along with requirements for using the `aria-*` attributes that supply the same [implicit ARIA semantics](#).

Rules of ARIA attribute usage by HTML feature

HTML feature	Implicit ARIA semantics	HTML feature and aria-* attribute author guidance
Any element where the checked attribute is allowed	aria-checked="true"	Use the <code>checked</code> attribute on any element that is allowed the <code>checked</code> attribute in HTML. Use the indeterminate IDL attribute to indicate the "mixed" state for input type=checkbox elements. Authors <i>MUST NOT</i> use the aria-checked attribute on any element where the checkedness, or the indeterminate checked value of the element can be in opposition to the current value of the <code>aria-checked</code> attribute. Authors <i>MAY</i> use the <code>aria-checked</code> attribute on any other element with a WAI-ARIA role which allows the attribute.
		Use the <code>disabled</code> attribute on any element that is allowed the <code>disabled</code> attribute in HTML. Authors <i>MAY</i> use the aria-disabled attribute on any element that is allowed the

Any element where the [disabled](#) attribute is allowed, including option [disabled](#) and optgroup [disabled](#)

aria-
disabled="true"

disabled attribute in HTML, or any element with a WAI-ARIA role which allows the aria-disabled attribute.

Authors *SHOULD NOT* use aria-disabled="true" on any element which also has a disabled attribute.

Authors *MUST NOT* use aria-disabled="false" on any element which also has a disabled attribute.

Authors *MAY* use the [aria-hidden](#) attribute on any HTML element that allows [global aria-* attributes](#) to be specified, with the exception of focusable elements and the [body](#) element.

It is generally *NOT RECOMMENDED* for authors to use aria-hidden="true" on any element which also has the hidden attribute specified. However, authors *MUST NOT* use aria-hidden="true" on any element which also has the hidden attribute specified in the until-found state.

NOTE

A focusable element is any element which can be focused by use of keyboard or pointer device. Focusable elements are not

Any element with a [hidden](#) attribute

`aria-hidden="true"`

always elements which can be tabbed to via a keyboard. For instance, an element with `tabindex="-1"` is focusable but is not a tabbable element.

NOTE

Using `aria-hidden="true"` on an element that has the `hidden` attribute is at best an unnecessary redundancy. At worst its usage can prevent access to the content if the `hidden` attribute's default UA style of `display: none` has been purposefully overwritten by an author or user style sheet. Finally, if the `hidden` attribute has the value of `until-found`, the use of `aria-hidden=true` will prevent this content from being discoverable to users of assistive technology when it is found via a browser's in-page find feature and visually rendered to users.

Use the `placeholder` attribute on any element that is allowed the `placeholder` attribute in HTML.

Any element where the placeholder attribute is allowed	aria- placeholder="..." Authors <i>MAY</i> use the aria-placeholder attribute on any element that is allowed the placeholder attribute in HTML, or any element with a WAI-ARIA role which allows the aria-placeholder attribute. Authors <i>MUST NOT</i> use the aria-placeholder attribute on any element which also has a placeholder attribute.
Any element where the max attribute is allowed: meter max, progress max, and input max	aria- valuemax="..." Use the max attribute on any element that is allowed the max attribute in HTML. Authors <i>MAY</i> use the aria-valuemax attribute on any other element with a WAI-ARIA role which allows the aria-valuemax attribute. Authors <i>SHOULD NOT</i> use aria-valuemax on any element which allows the max attribute. Use the max attribute instead. Authors <i>MUST NOT</i> use aria-valuemax on any element which also has a max attribute.
	Use the min attribute on any element that is allowed the min attribute in HTML. Authors <i>MAY</i> use the aria-valuemin attribute on any other

Any element where the <code>min</code> attribute is allowed: <code>meter</code> min and <code>input</code> min	<code>aria-valuemin="..."</code>	element with a WAI-ARIA role which allows the <code>aria-valuemin</code> attribute. Authors <i>SHOULD NOT</i> use <code>aria-valuemin</code> on any element which allows the <code>min</code> attribute. Use the <code>min</code> attribute instead. Authors <i>MUST NOT</i> use <code>aria-valuemin</code> on any element which also has a <code>min</code> attribute.
Any element which allows the <code>readonly</code> attribute: <code>input</code> readonly, <code>textarea</code> readonly and form-associated custom element which allows readonly	<code>aria-readonly="true"</code>	Use the <code>readonly</code> attribute on any element that is allowed the <code>readonly</code> attribute in HTML. Authors <i>MAY</i> use the aria-readonly attribute on any element with a WAI-ARIA role which allows the attribute. Authors <i>SHOULD NOT</i> use the <code>aria-readonly="true"</code> on any element which also has a <code>readonly</code> attribute. Authors <i>MUST NOT</i> use <code>aria-readonly="false"</code> on any element which also has a <code>readonly</code> attribute.
Element with contenteditable=true or element without <code>contenteditable</code> attribute whose closest ancestor with a <code>contenteditable</code> attribute has		Authors <i>MUST NOT</i> set <code>aria-</code>

contenteditable="true". NOTE This is equivalent to the isContentEditable IDL attribute.	aria-readonly="false" readonly="true" on an element that has isContentEditable="true".
Any element where the required attribute is allowed: input required, textarea required, and select required	aria-required="true" Use the required attribute on any element that is allowed the required attribute in HTML. Authors <i>MAY</i> use the aria-required attribute on any element that is allowed the required attribute in HTML, or any element with a WAI-ARIA role which allows the aria-required attribute. Authors <i>SHOULD NOT</i> use the aria-required="true" on any element which also has a required attribute. Authors <i>MUST NOT</i> use aria-required="false" on any element which also has a required attribute.
Any element where the	aria- Use the colspan attribute on any element that is allowed the colspan attribute in HTML. Authors <i>SHOULD NOT</i> use the aria-colspan attribute on any element which also has a

colspan attribute is allowed: <code>td</code> and <code>th</code>	<code>colspan="..."</code>	element which also has a <code>colspan</code> attribute. Authors <i>MUST NOT</i> use <code>aria-colspan</code> on any element which also has a <code>colspan</code> attribute, and the values of each attribute do not match.
Any element where the rowspan attribute is allowed: <code>td</code> and <code>th</code>	<code>aria-rowspan="..."</code>	Use the <code>rowspan</code> attribute on any element that is allowed the <code>rowspan</code> attribute in HTML. Authors <i>SHOULD NOT</i> use the <code>aria-rowspan</code> attribute on any element which also has a <code>rowspan</code> attribute. Authors <i>MUST NOT</i> use <code>aria-rowspan</code> on any element which also has a <code>rowspan</code> attribute, and the values of each attribute do not match.

4.3 Requirements for deprecated ARIA role, state and property and attributes

The ARIA Specification's [Deprecated Requirements](#) section indicates that if an ARIA feature is marked as deprecated then authors are advised not to use said feature for new content.

The following roles and attributes are deprecated features of ARIA and DPub ARIA. Conformance checkers *MUST* warn authors about the deprecated status of these features. Whenever possible, authors are advised to use alternatives to deprecated features.

4.3.1 Deprecated ARIA roles

- [directory](#)

NOTE

The `directory` role is marked for deprecation in [WAI-ARIA 1.2](#). In reality, the `directory` role had no functional difference to an element with an implicit or explicit `list` role. Authors are advised to use one of HTML's native list elements, or an ARIA `list` instead.

4.3.2 Deprecated DPub ARIA roles

- [doc-biblioentry](#)
- [doc-endnote](#)

NOTE

The `doc-biblioentry` and `doc-endnote` roles are marked for deprecation in [Digital Publishing WAI-ARIA Module 1.1](#), as they are not valid children for an element with an implicit or explicit role of `list`. Authors can use standard list and child `li` elements without the need for these roles.

4.3.3 Deprecated ARIA attributes

- [aria-dropeffect](#)
- [aria-grabbed](#)

NOTE

The `aria-dropeffect` and `aria-grabbed` attributes were deprecated in [WAI-ARIA 1.1](#). There is presently no feature in ARIA to replace their proposed functionality.

§ 4.4 Case requirements for ARIA role, state and property attributes

Authors *SHOULD* use [ASCII lowercase](#) for all `role` token values and any state or property attributes (`aria-*`) whose values are [defined as tokens](#).

NOTE

While modern browsers treat the `role` or `aria-*` attribute values as [ASCII case-insensitive](#), not all assistive technologies will correctly parse these values.

To reduce interoperability issues, authors are strongly encouraged to use [ASCII lowercase](#) for `aria-*` and `role` attribute values. Further, authors are encouraged to rigorously test with different browser and assistive technology combinations to ensure that their content will be correctly exposed to their users.

EXAMPLE 16

Correct (conforming) examples:

```
<div role="main">...</div>
```

```
<a href="home/" aria-current="page">home</a>
```

Incorrect (non-conforming) examples:

```
<!-- DO NOT DO THE FOLLOWING -->
```

```
<div role="MAIN">...</div>
```

```
<div role="Main">...</div>
```

```
<a href="home/" aria-current="Page">home</a>
```

§ 5. Allowed descendants of ARIA roles

This section is non-normative.

The following table maps (and extends) the [Kinds of content](#) and allowed descendant information (defined in the [\[HTML\]](#) specification) to elements that have an equivalent role.

Column 1 links to the normative [Accessible Rich Internet Applications \(WAI-ARIA\) 1.2](#) definitions for each ARIA role. Column 2 identifies the [Kinds of content](#) categories each role has when it is used on an HTML element. Column 3 indicates what kinds of HTML elements can be descendants of an element with an explicit role specified, often matching the HTML element with the same [implicit](#) role.

For example, a [button](#) element has an implicit role=button. In HTML a button element allows [phrasing content](#) as descendants, and does not allow [interactive content](#) or descendants with a `tabindex` attribute. Therefore, any elements specified with a `role=button` would follow the same descendant restrictions, and not allow any interactive content descendants, elements with a `tabindex` specified, or any elements with role values that are in the interactive content category (identified in column 3).

Examples of non-conforming descendants

```
<!-- conformance checkers will report an error -->
<button>
  <div role="button">...</div>
</button>

<div role="button">
  <button>...</button>
</div>

<div role="link">
  <textarea>...</textarea>
</div>
```

Additionally, there are certain roles which [Accessible Rich Internet Applications \(WAI-ARIA\) 1.2](#) has specified specific requirements for their allowed descendants. These have been identified in column 3 (Descendant allowances) by indicating to "Refer to the 'Required Owned Elements'" for those particular roles.

Allowed descendants of ARIA roles

Role	Kind of content	Descendant allowances
<u>alert</u>	<u>Flow content</u>	<u>Flow content</u> but with no <u>main</u> element descendants.
<u>alertdialog</u>	<u>Flow content</u>	<u>Flow content</u>
<u>application</u>	<u>Flow content</u>	<u>Flow content</u>
<u>article</u>	• <u>Flow content</u> • <u>Sectioning content</u> • <u>Palpable content</u>	<u>Flow content</u> but with no <u>main</u> element descendants.
<u>banner</u>	• <u>Flow content</u> • <u>Palpable content</u>	<u>Flow content</u> but with no <u>main</u> , <u>header</u> , or <u>footer</u> element descendants.
<u>blockquote</u>	• <u>Flow content</u> • <u>Palpable content</u>	<u>Flow content</u> but with no <u>main</u> element descendants.
<u>button</u>	• <u>Flow content</u> • <u>Phrasing content</u> • <u>Interactive content</u> • <u>Palpable content</u>	<u>Phrasing content</u> , but with no <u>interactive content</u> descendants, and no descendants with a <code>tabindex</code> attribute specified.
<u>caption</u>	N/A	<u>Flow content</u> but with no <u>main</u> or <u>table</u> element descendants.
<u>cell</u>	N/A	<u>Flow content</u> but with no <u>main</u> element descendants.

<u>checkbox</u>	• Flow content • Phrasing content • Interactive content	Phrasing content , but with no interactive content descendants, and no descendants with a <code>tabindex</code> attribute specified.
<u>code</u>	• Flow content • Phrasing content • Palpable content	Phrasing content
<u>columnheader</u>	N/A	Flow content but with no main , header , or footer element descendants.
<u>combobox</u>	• Flow content • Phrasing content • Interactive content • Palpable content	Flow content but with no main element descendants.
<u>complementary</u>	• Flow content • Sectioning content • Palpable content	Flow content but with no main element descendants.
<u>contentinfo</u>	• Flow content • Palpable content	Flow content but with no main , header , or footer element descendants.
<u>definition</u>	• Flow content • Phrasing content	Phrasing content

<u>definition</u>	• <u>Phrasing content</u>	<u>Phrasing content</u>
	• <u>Palpable content</u>	
<u>deletion</u>	• <u>Flow content</u> • <u>Phrasing content</u>	<u>Phrasing content</u>
<u>dialog</u>	<u>Flow content</u>	<u>Flow content</u>
<u>directory</u>	<u>Flow content</u>	<u>Flow content</u> but with no <u>main</u> element descendants.
<u>document</u>	<u>Flow content</u>	<u>Flow content</u>
<u>emphasis</u>	• <u>Flow content</u> • <u>Phrasing content</u> • <u>Palpable content</u>	<u>Phrasing content</u>
<u>feed</u>	<u>Flow content</u>	<u>Flow content</u> but with no <u>main</u> element descendants.
<u>figure</u>	• <u>Flow content</u> • <u>Palpable content</u>	<u>Flow content</u> but with no <u>main</u> element descendants.
<u>form</u>	• <u>Flow content</u> • <u>Palpable content</u>	<u>Flow content</u> , but with no <u>form</u> element descendants.
<u>generic</u>	• <u>Flow content</u> • <u>Phrasing content</u> • <u>Palpable content</u>	<u>Flow content</u>
	• <u>Flow content</u>	

<u>grid</u>	• Interactive content • Palpable content	Refer to the "Required Owned Elements" as defined for the ARIA grid role.
<u>gridcell</u>	Interactive content	Flow content but with no main element descendants.
<u>group</u>	• Flow content • Palpable content	Flow content
<u>heading</u>	• Flow content • Heading content • Palpable content	Phrasing content
<u>img or image</u>	• Flow content • Phrasing content • Embedded content • Palpable content	Phrasing content , but with no interactive content descendants.
<u>insertion</u>	• Flow content • Phrasing content • Palpable content	Phrasing content
<u>link</u>	• Flow content • Phrasing content • Interactive content	Flow content , but with no interactive content descendants, and no descendants with a <code>tabindex</code> attribute specified.

	• Palpable content	
list	Flow content	Refer to the "Required Owned Elements" as defined for the ARIA list role.
listbox	• Flow content • Interactive content • Palpable content	Refer to the "Required Owned Elements" as defined for the ARIA listbox role.
listitem	N/A	Flow content but with no main element descendants.
log	Flow content	Flow content , but with no main element descendants.
main	Flow content	Flow content , but with no main element descendants.
marquee	Flow content	Flow content , but with no main element descendants.
math	Flow content	Flow content
menu	• Flow content • Interactive content	Refer to the "Required Owned Elements" as defined for the ARIA menu role.
menubar	• Flow content • Interactive content	Refer to the "Required Owned Elements" as defined for the ARIA menubar role.
menuitem	Interactive content	Phrasing content , but with no interactive content descendants, and no descendants with a <code>tabindex</code> attribute specified.

<u>menuitemcheckbox</u>	<u>Interactive content</u>	<u>Phrasing content</u> , but with no <u>interactive content</u> descendants, and no descendants with a <code>tabindex</code> attribute specified.
<u>menuitemradio</u>	<u>Interactive content</u>	<u>Phrasing content</u> , but with no <u>interactive content</u> descendants, and no descendants with a <code>tabindex</code> attribute specified.
<u>meter</u>	• <u>Flow content</u> • <u>Phrasing content</u> • <u>Palpable content</u>	<u>Phrasing content</u> , but with no <u>meter</u> element descendants.
<u>navigation</u>	• <u>Flow content</u> • <u>Sectioning content</u> • <u>Palpable content</u>	<u>Flow content</u> , but with no <u>main</u> element descendants.
<u>none</u>	N/A	<u>Transparent</u>
<u>note</u>	<u>Flow content</u>	<u>Flow content</u> , but with no <u>main</u> element descendants.
<u>option</u>	<u>Interactive content</u>	<u>Phrasing content</u> , but with no <u>interactive content</u> descendants, and no descendants with a <code>tabindex</code> attribute specified.
<u>paragraph</u>	• <u>Flow content</u> • <u>Palpable content</u>	<u>Phrasing content</u>
<u>presentation</u>	N/A	<u>Transparent</u>
<u>progressbar</u>	• <u>Flow content</u> • <u>Phrasing content</u>	<u>Phrasing content</u> , but with no <u>progress</u> element descendants.

<u>radio</u>	• <u>Flow content</u> • <u>Phrasing content</u> • <u>Interactive content</u>	<u>Phrasing content</u> , but with no <u>interactive content</u> descendants, and no descendants with a <code>tabindex</code> attribute specified.
<u>radiogroup</u>	• <u>Flow content</u> • <u>Palpable content</u>	<u>Flow content</u>
<u>region</u>	• <u>Flow content</u> • <u>Sectioning content</u> • <u>Palpable content</u>	<u>Flow content</u> , but with no <u>main</u> element descendants.
<u>row</u>	N/A	Refer to the "Required Owned Elements" as defined for the ARIA <u>row</u> role.
<u>rowgroup</u>	N/A	Refer to the "Required Owned Elements" as defined for the ARIA <u>rowgroup</u> role.
<u>rowheader</u>	N/A	<u>Flow content</u> but with no <u>main</u> element descendants.
<u>scrollbar</u>	<u>Interactive content</u>	<u>Phrasing content</u>
<u>search</u>	• <u>Flow content</u> • <u>Palpable content</u>	<u>Flow content</u> but with no <u>main</u> element descendants.
<u>searchbox</u>	• <u>Flow content</u> • <u>Phrasing content</u> • <u>Interactive content</u>	<u>Flow content</u> but with no <u>main</u> element descendants.

<u>separator</u>	Interactive content (if focusable)	Phrasing content
<u>slider</u>	• Flow content • Phrasing content • Interactive content	Phrasing content
<u>spinbutton</u>	• Flow content • Phrasing content • Interactive content	Flow content but with no main element descendants.
<u>status</u>	Flow content	Flow content but with no main element descendants.
<u>strong</u>	• Flow content • Phrasing content • Palpable content	Phrasing content
<u>subscript</u>	• Flow content • Phrasing content • Palpable content	Phrasing content
<u>superscript</u>	• Flow content • Phrasing content • Palpable content	Phrasing content
	• Flow content	

<u>switch</u>	• <u>Phrasing content</u> • <u>Interactive content</u>	<u>Phrasing content</u> , but with no <u>interactive content</u> descendants, and no descendants with a <code>tabindex</code> attribute specified.
<u>tab</u>	<u>Interactive content</u>	<u>Phrasing content</u> , but with no <u>interactive content</u> descendants, and no descendants with a <code>tabindex</code> attribute specified.
<u>table</u>	• <u>Flow content</u> • <u>Palpable content</u>	Refer to the "Required Owned Elements" as defined for the ARIA <u>table</u> role.
<u>tablist</u>	• <u>Flow content</u> • <u>Palpable content</u>	Refer to the "Required Owned Elements" as defined for the ARIA <u>tablist</u> role.
<u>tabpanel</u>	<u>Flow content</u>	<u>Flow content</u>
<u>term</u>	<u>Phrasing content</u>	<u>Phrasing content</u>
<u>textbox</u>	<u>Interactive content</u>	<u>Flow content</u> but with no <u>main</u> element descendants.
<u>time</u>	• <u>Flow content</u> • <u>Phrasing content</u> • <u>Palpable content</u>	<u>Phrasing content</u>
<u>timer</u>	<u>Flow content</u>	<u>Flow content</u> but with no <u>main</u> element descendants.
<u>toolbar</u>	<u>Flow content</u>	<u>Flow content</u> but with no <u>main</u> element descendants.
<u>tooltip</u>	<u>Flow content</u>	<u>Phrasing content</u>
<u>tree</u>	<u>Flow content</u>	Refer to the "Required Owned Elements" as defined for the ARIA <u>tree</u> role.

treegrid	Flow content	Refer to the "Required Owned Elements" as defined for the ARIA treegrid role.
treeitem	Interactive content	Phrasing content

§ 6. Conformance

As well as sections marked as non-normative, all authoring guidelines, diagrams, examples, and notes in this specification are non-normative. Everything else in this specification is normative.

The key words *MAY*, *MUST*, *MUST NOT*, *NOT RECOMMENDED*, *SHOULD*, and *SHOULD NOT* in this document are to be interpreted as described in [BCP 14](#) [*RFC2119*] [*RFC8174*] when, and only when, they appear in all capitals, as shown here.

§ 6.1 Conformance checking requirements

Conformance checkers that claim support for checking ARIA in HTML documents *MUST* implement checks for the conformance requirements for use of the ARIA `role` and `aria-*` attributes on [HTML elements](#) as defined in this specification.

A conforming document *MUST NOT* contain any elements with author defined `role` or `aria-*` attributes with values other than those which, per this specification, authors *MAY* use on each [HTML element](#) in [4. Document conformance requirements for use of ARIA attributes in HTML](#).

Conformance checkers *SHOULD* flag instances where authors are explicitly providing an element with a `role` which matches its [implicit ARIA semantics](#) as failures, as it is *NOT RECOMMENDED* for authors to explicitly set these roles.

A conformance checker *MAY* define their own terminology, and level or levels of severity, when surfacing document failures to conform to this specification.

§ 7. Privacy and security considerations

This section is non-normative.

This specification does not define the features of [*wai-aria-1.2*], [*dpub-aria-1.0*] or [*HTML*]. Rather it provides rules and guidance for conformance checkers that claim support for checking ARIA in HTML, as well as providing guidance to authors.

Therefore, there are no known privacy or security impacts of this specification, as it defines no new features to introduce potential concern.

§ A. References

§ A.1 Normative references

[accname-1.2]

Accessible Name and Description Computation 1.2. Bryan Garaventa; Melanie Sumner. W3C. 17 June 2025. W3C Working Draft. URL: <https://www.w3.org/TR/accname-1.2/>

[dpub-aria-1.0]

Digital Publishing WAI-ARIA Module 1.0. Matt Garrish; Tzviya Siegman; Markus Gylling; Shane McCarron. W3C. 14 December 2017. W3C Recommendation. URL: <https://www.w3.org/TR/dpub-aria-1.0/>

[dpub-aria-1.1]

Digital Publishing WAI-ARIA Module 1.1. Matt Garrish; Tzviya Siegman. W3C. 12 June 2025. W3C Recommendation. URL: <https://www.w3.org/TR/dpub-aria-1.1/>

[html]

HTML Standard. Anne van Kesteren; Domenic Denicola; Dominic Farolino; Ian Hickson; Philip Jägenstedt; Simon Pieters. WHATWG. Living Standard. URL: <https://html.spec.whatwg.org/multipage/>

[html-aam-1.0]

HTML Accessibility API Mappings 1.0. Scott O'Hara; Rahim Abdi. W3C. 30 July 2025. W3C Working Draft. URL: <https://www.w3.org/TR/html-aam-1.0/>

[infra]

Infra Standard. Anne van Kesteren; Domenic Denicola. WHATWG. Living Standard. URL: <https://infra.spec.whatwg.org/>

[RFC2119]

Key words for use in RFCs to Indicate Requirement Levels. S. Bradner. IETF. March 1997. Best

Current Practice. URL: <https://www.rfc-editor.org/rfc/rfc2119>

[RFC8174]

[*Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words*](#). B. Leiba. IETF. May 2017. Best Current Practice. URL: <https://www.rfc-editor.org/rfc/rfc8174>

[svg-aam-1.0]

[*SVG Accessibility API Mappings*](#). Amelia Bellamy-Royds; Ian Pouncey. W3C. 10 May 2018. W3C Working Draft. URL: <https://www.w3.org/TR/svg-aam-1.0/>

[wai-aria-1.1]

[*Accessible Rich Internet Applications \(WAI-ARIA\) 1.1*](#). Joanmarie Diggs; Shane McCarron; Michael Cooper; Richard Schwerdtfeger; James Craig. W3C. 14 December 2017. W3C Recommendation. URL: <https://www.w3.org/TR/wai-aria-1.1/>

[wai-aria-1.2]

[*Accessible Rich Internet Applications \(WAI-ARIA\) 1.2*](#). Joanmarie Diggs; James Nurthen; Michael Cooper; Carolyn MacLeod. W3C. 6 June 2023. W3C Recommendation. URL: <https://www.w3.org/TR/wai-aria-1.2/>

§ A.2 Informative references

[using-aria]

[*Using ARIA*](#). Steve Faulkner; David MacDonald. W3C. 27 September 2018. W3C Working Draft. URL: <https://www.w3.org/TR/using-aria/>

[wai-aria-practices-1.2]

[*WAI-ARIA Authoring Practices 1.2*](#). Matthew King; JaEun Jemma Ku; James Nurthen; Zoë Bijl; Michael Cooper. W3C. 19 May 2022. W3C Working Group Note. URL: <https://www.w3.org/TR/wai-aria-practices-1.2/>

↑