

ARIA Serious?!

Eric Eggert

Eric Eggert is a Web Developer and Teacher who works with Knowbility and the W3C on improving the Web for People with Disabilities, and everyone else.



Knowbility

W3C®

Important Note:

This presentation contains ARIA examples that are preventing websites and applications from being accessible.

Don't just copy & paste code.

WAI-ARIA: Accessible Rich Internet Applications

Internet Applications

Internet Applications

ARIA ≠



Or



No
aria-make-accessible="true"



Rules of ARIA¹

¹ [Using ARIA](#)

If you **can** use a native
HTML element or
attribute with the
semantics and
behavior you require
already built in, *instead*
of re-purposing an
element and adding
an ARIA role, state or
property to make it
accessible, then do so.

*Do not change
native
semantics,
unless you really
have to.*



Bad Example:

```
<h1 role="button">heading button</h1>
```

Good Example:

```
<h1><button>heading button</button></h1>
```



Nicolas Steenhout

@vavroom



Here's an accessibility gem: <h2 class="h3" role="heading" aria-level="1">Some heading</h2> [#a11y](#)

36 12:48 PM - Oct 19, 2017



[21 people are talking about this](#)



*All interactive
ARIA controls
must be usable
with the
keyboard.*





If you create a widget that a user can click or tap or drag or drop or slide or scroll, a user must also be able to navigate to the widget and perform an equivalent action using the keyboard.



All interactive widgets must be scripted to respond to standard keystrokes or keystroke combinations where applicable.

Example

If using `role=button` the element must be able to receive focus and a user must be able to activate the action associated with the element using **both** the **enter** (on WIN OS) or **return** (MAC OS) and the **space key**.

— *ARIA Practices Guide*

DEMO

Open in Overcast y2k T LEO YT EPG W3C knowbility outline misc Snooker.org Oh, shit, git! LiquidCheatsheet uberspace.de Forms Tables a1y.css Target Size Target Size My Bookmarklet Lists Bookmarklet Text Spacing >> +

codepen.io/yatil/pen/WBRBpP

Heart Save Fork Settings Change View

Button vs div

A PEN BY Eric Eggert PRO

HTML

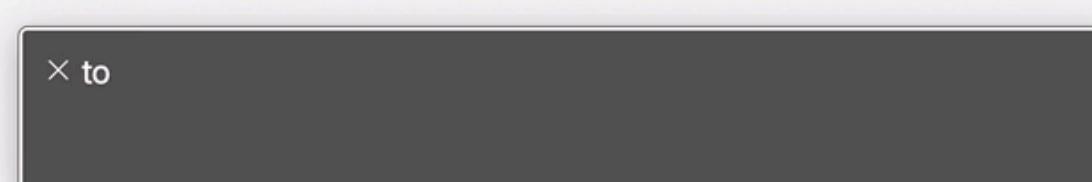
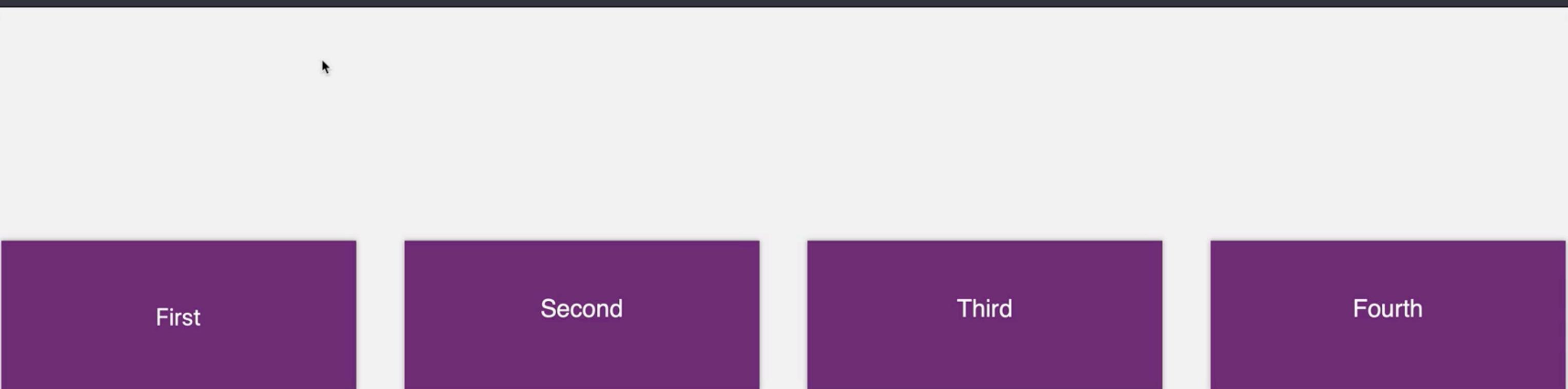
```
4 <p>Compare behavior of the following elements. Try clicking on them with your  
mouse. Then use the keyboard's TAB key or SHIFT+TAB to navigate forward or  
back. Trigger the <em>buttons</em> using either the ENTER key or spacebar.  
</p>  
5 <h2>Buttons and divs</h2>-->
```

CSS

```
1 body {  
2   display: flex;  
3   justify-content: center;  
4   font-family: sans-serif;  
5   height: 100%;  
...
```

JS

```
1 var btns = document.querySelectorAll('.bouton');  
2  
3 for(var i = 0; i < btns.length; i++) {  
4   btns[i].addEventListener('click', function(e) {  
5     alert('You clicked, the dialog box opened.');
```



***Do not use
role="presentation"ⁱ or
aria-hidden="true" on a
focusable element.***

***Using either of these
on a focusable element
will result in some users
focusing on 'nothing'.***

ⁱwhich is equivalent to role="none"

*All interactive
elements must
have an
accessible
name.*



ARIA Components

Some Roles

```
<div role="main"> ← better: <main> →  
<form role="search">  
<div role="tooltip">  
<output role="alert">
```

Some Properties and States

```
<input aria-required="true">
```

```
<input aria-labelledby="label" aria-describedby="errormsg">
```

```
<button aria-pressed="false">
```

```
<div aria-hidden="true">
```

Live Regions

```
<div aria-live="off">  
  
<div aria-live="polite">  
  
<div aria-live="assertive">
```

ARIA Support

Heydon Pickering:
aria-controls is 

↪ hey don't Retweeted



Léonie

@LeonieWatson

Replying to @heydonworks

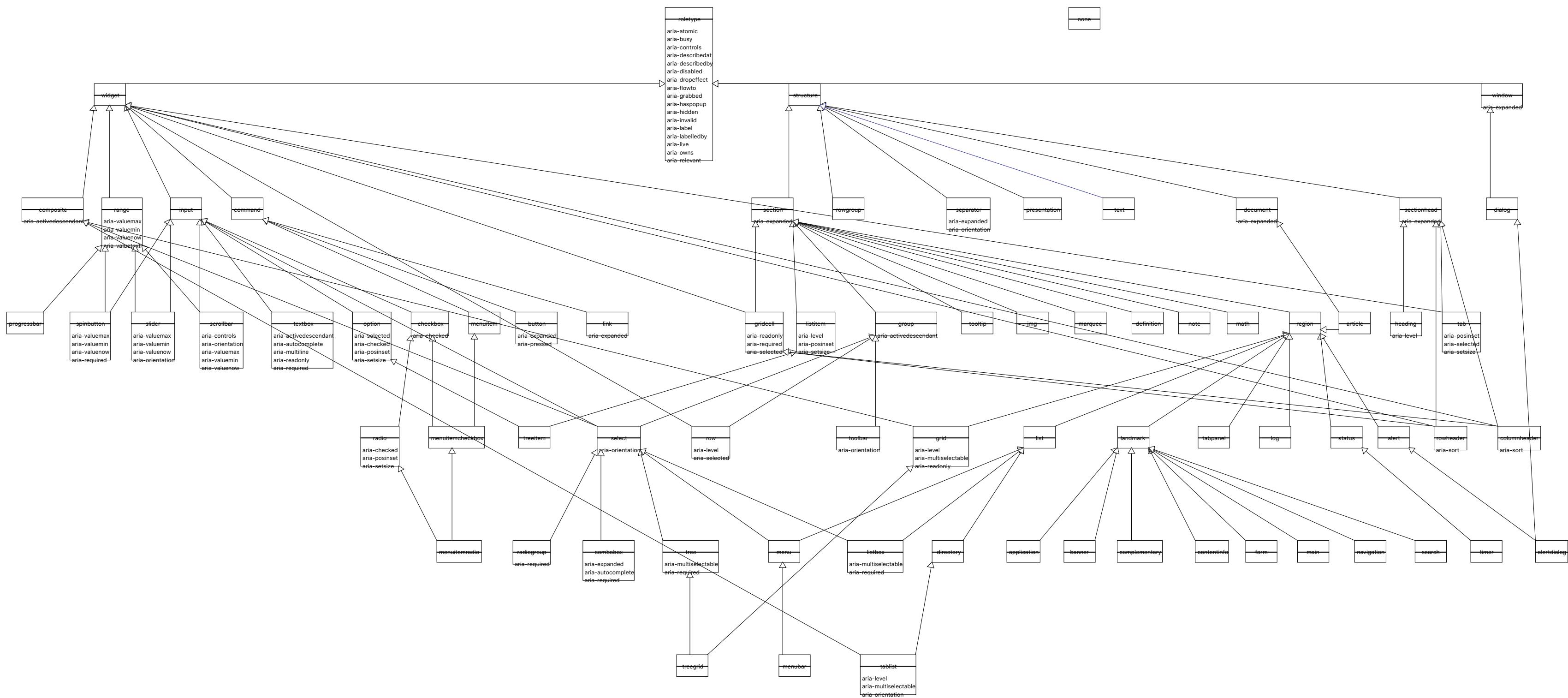
Jaws has recently reduced support, so it now has to be enabled. A real shame, since -controls is the only solution to a valid use case IMO. @katannshaw @FreedomSci
@AppleSupport @NVAccess @MSFTEnable @gnome

1:52 PM · Oct 2, 2019 · OpenTweet

3 Retweets 2 Likes

Léonie Watson on Twitter, October 2, 2019





WAI ARIA is really
Complicated!

Makes it also complicated
for Web Developers!

Exhibit 1 – Bad!

```
<a id="airline-logo" href="..."  
  class="logo"  
  aria-label="Airline Name">  
  &nbsp;  
</a>
```

Exhibit 1 – Better!

```
<a id="logo" href="#">  
    
</a>
```

or:

```
<a id="logo" href="#">  
  <svg>  
    <title>Airline Name</title>  
    ...  
  </svg>  
</a>
```

Exhibit 2 – Bad!

```
<div class="nav">
  <a href="javascript:void(0);" class="navInactive" role="button">
    <span class="hiddenText">Slide 1</span>
  </a>
  <a href="javascript:void(0);" class="navActive" role="button">
    <span class="hiddenText">Slide 2</span>
  </a>
  <a href="javascript:void(0);" class="navInactive" role="button">
    <span class="hiddenText">Slide 3</span>
  </a>
</div>
```

Exhibit 2 – Better!

```
<nav>
  <button aria-pressed="false" aria-label="Slide 1">
    &nbsp;
  </button>
  <button aria-pressed="true" aria-label="Slide 2">
    &nbsp;
  </button>
  <button aria-pressed="false" aria-label="Slide 3">
    &nbsp;
  </button>
</nav>
```

Exhibit 2 – Even Better!

```
<nav>
  <button aria-pressed="false">
    
  </button>
  <button aria-pressed="true">
    
  </button>
  <button aria-pressed="false">
    
  </button>
</nav>
```

Exhibit 3 – Bad!

```
<th  
tabindex="0"  
role="button"  
aria-label="Sort column"  
>Name</th>
```

Exhibit 3a – Bad!



Paul J. Adam
@pauljadam

<button type="button" role="button" aria-required="false" tabindex="0">2</button> WHAT THE HECK? [#facepalm](#)
[#a11y](#)

5:36 PM · Jan 10, 2018 · Twitter for Mac

3 Retweets 13 Likes

Comment Reply Like Share

Exhibit 4 – Bad!²

```
<span  
aria-hidden="true"  
role="img"  
class="icon">  
</span>
```

²Actually not harmful, but really, what is the point?!

Exhibit 5 – Really Bad!

```
<body aria-hidden="true">
```

Exhibit 6 – Bad!

```
<a  
    href="..."  
    target="_blank"  
    title="Click here to  
        view the video."  
    tabindex="-1"  
    role="button"  
    aria-label="External link"  
></a>
```



Eric Eggert

@yatil

▼

Things you should not do: use an aria-live region on a carousel. Screen readers get constantly disrupted because the content changes. Makes your website impossible to use.

#ariaserious

3:39 AM · May 8, 2018 · Twitterrific for Mac



Eric Eggert
@yatil

▼

```
<label for="ID">Label Text</label>
<input type="text" id="id" />
```

Is not accessible. IDs are case sensitive in HTML/the DOM.

#ariaserious #htmlserious

5:17 AM · May 8, 2018 · Twitterrific for Mac

TABLE OF CONTENTS

1. Introduction
2. Read Me First
 - 2.1 No ARIA is better than Bad ARIA
 - 2.2 Browser and Assistive Technology Support
 - 2.3 Mobile and Touch Support
3. Design Patterns and Widgets
 - 3.1 Accordion (Sections With Show/Hide Functionality)
 - 3.2 Alert
 - 3.3 Alert and Message Dialogs
 - 3.4 Breadcrumb
 - 3.5 Button
 - 3.6 Carousel (Slide Show or Image Rotator)
 - 3.7 Checkbox
 - 3.8 Combo Box
 - 3.9 Dialog (Modal)
 - 3.10 Disclosure (Show/Hide)
 - 3.11 Feed
 - 3.12 Grids : Interactive Tabular Data and Layout Containers
 - 3.13 Link
 - 3.14 Listbox
 - 3.15 Menu or Menu bar
 - 3.16 Menu Button
 - 3.17 Radio Group
 - 3.18 Slider
 - 3.19 Slider (Multi-Thumb)
 - 3.20 Spinbutton
 - 3.21 Table
 - 3.22 Tabs
 - 3.23 Toolbar
 - 3.24 Tooltip Widget
 - 3.25 To... Vi...

WAI-ARIA Authoring Practices 1.1

W3C Working Group Note 07 February 2019

**This version:**

<https://www.w3.org/TR/2019/NOTE-wai-aria-practices-1.1-20190207/>

Latest published version:

<https://www.w3.org/TR/wai-aria-practices-1.1/>

Latest editor's draft:

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

Previous version:

<https://www.w3.org/TR/2018/NOTE-wai-aria-practices-1.1-20180726/>

Editors:

[Matt King](#) (Facebook)

[JaEun Jemma Ku](#) (University of Illinois)

[James Nurthen](#) (Adobe)

Michiel Bijl (Invited Expert)

[Michael Cooper](#) (W3C)

Former editors:

Joseph Scheuhammer (Inclusive Design Research Centre, OCAD University) (Editor until October 2014)

Lisa Pappas (SAS) (Editor until October 2009)

Rich Schwerdtfeger (IBM Corporation) (Editor until October 2014)

[Copyright](#) © 2015-2019 W3C® ([MIT](#), [ERCIM](#), [Keio](#), [Beihang](#)). W3C [liability](#), [trademark](#) and [permissive document license](#) rules apply.

Abstract

This document provides readers with an understanding of how to use [WAI-ARIA 1.1](#) [[WAI-ARIA](#)] to create accessible rich internet applications. It describes considerations that might not be evident to most authors from the [WAI-ARIA](#) specification alone and recommends approaches to make widgets, navigation, and behaviors accessible using [WAI-ARIA](#) roles, states, and properties. This document is directed primarily to Web application developers, but the guidance is also useful for user agent and assistive technology developers.

This document is part of the [WAI-ARIA](#) suite described in the [WAI-ARIA Overview](#).

Status of This Document

This section describes the status of this document at the time of its publication. Other documents may supersede this document. A list of current W3C publications and the latest revision of this technical report can be found in the [W3C technical reports index](#) at <https://www.w3.org/TR/>.



TABLE OF CONTENTS

1. [Introduction](#)
2. [Notes on ARIA Use in HTML](#)
 - 2.1 First Rule of ARIA Use
 - 2.2 Second Rule of ARIA Use
 - 2.3 Third Rule of ARIA Use
 - 2.4 Fourth Rule of ARIA Use
 - 2.5 Fifth Rule of ARIA Use
 - 2.6 What Does Adding a Role Do to the Native Semantics?
 - 2.7 Add ARIA Inline or via Script?
 - 2.8 ARIA Validation
 - 2.9 Use of Role=presentation or Role=none
 - 2.10 Practical Support: aria-label, aria-labelledby and aria-describedby
 - 2.10.1 The Effect of Accessible Name on Background Images
 - 2.11 Using ARIA role=application
 - 2.12 Custom Control Accessible Development Checklist:
 - 2.13 ARIA Adds Nothing to Default Semantics of Most HTML Elements
 - 2.13.1 Some Examples of Redundant ARIA
 - 2.14 Aria Roles and Properties Not Available as Features in HTML
 - 2.14.1 ARIA Roles
 - 2.14.2 ARIA States and Properties (aria-* Attributes)
 - 2.15 ARIA Design Patterns and Touch Device Support
 - 2.16 Recommendations Table:
 - 2.17 ARIA Role, State, and Property Quick Reference
- A. [References](#)
 - A.1 Informative references

Using ARIA

W3C Working Draft 27 September 2018



This version:

<https://www.w3.org/TR/2018/WD-using-aria-20180927/>

Latest published version:

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

Latest editor's draft:

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

Previous version:

<https://www.w3.org/TR/2018/WD-using-aria-20180924/>

Editors:

[Steve Faulkner](#) (The Paciello Group)

[David MacDonald](#) (CanAdapt Solutions Inc.)

Participate:

[GitHub w3c/using-aria](#)

[File a bug](#)

[Commit history](#)

[Pull requests](#)

Copyright © 2018 W3C® ([MIT](#), [ERCIM](#), [Keio](#), [Beihang](#)). W3C [liability](#), [trademark](#) and [permissive document license](#) rules apply.

Abstract

This document is a practical guide for developers on how to add accessibility information to HTML elements using the Accessible Rich Internet Applications specification [[WAI-ARIA-1.1](#)], which defines a way to make Web content and Web applications more accessible to people with disabilities. This document demonstrates how to use ARIA in [[HTML51](#)], which especially helps with dynamic content and advanced user interface controls developed with Ajax, HTML, JavaScript, and related technologies.

Status of This Document

This section describes the status of this document at the time of its publication. Other documents may supersede this document. A list of current W3C publications and the latest revision of this technical report can be found in



Conclusion

- ✓ Landmarks
- ✓ States & Properties
- ⚠ (Widget) Roles —
Mind the Keyboard

Workshop in Berlin

in November

Eric Eggert

yatil.net / @yatil everywhere